

UNIVERSITÉ DU QUÉBEC

THÈSE PRÉSENTÉE À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DU DOCTORAT EN GÉNIE ÉLECTRIQUE

PAR
QUOC-THAI HO

ARCHITECTURES VLSI DES RÉCEPTEURS À USAGERS MULTIPLES POUR LA
COMMUNICATION SANS FIL DE TROISIÈME GÉNÉRATION WCDMA

MAI 2008

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

© QUOC-THAI HO 2008

RÉSUMÉ

Les interférences – interférences inter-symboles (ISI) et interférences à accès multiples (MAI) – s'avèrent problématiques pour la reconstruction des données dans les systèmes de communications sans fil de troisième génération (3G). Ainsi, étant donné la présence des MAI, la détection à usagers multiples (MUD) devient une partie essentielle dans le déploiement des systèmes 3G au niveau des récepteurs. La MUD constitue un sujet de recherche continue depuis plus de vingt ans. L'aspect algorithmique de la MUD est bien étudié et documenté. Il existe plusieurs algorithmes sophistiqués permettant d'annuler efficacement des MAI. Toutefois, à cause de la complexité d'implémentation tout en satisfaisant des performances algorithmiques, l'aspect d'implémentation VLSI reste encore non-trivial. Cette thèse porte donc sur l'implémentation VLSI et la méthodologie de conception des récepteurs à usagers multiples.

La contribution majeure de ce travail doctoral consiste à proposer et à développer les architectures VLSI parallèles d'une classe d'algorithmes adaptatifs de MUD. Ces architectures satisfont les performances algorithmiques pour plusieurs débits de données dans plusieurs scénarios de communications WCDMA. Aussi, le récepteur étudié a été étendu et implémenté pour les systèmes multi-débit de données. Tenant compte des limitations des outils de conception commerciaux, et des limitations des méthodologies de conception à l'heure actuelle, la deuxième contribution de ce travail porte sur les méthodologies de conception. Il s'agit de deux méthodologies de conception clairement établies des systèmes complexes en utilisant les outils commerciaux. La première, l'approche conventionnelle, concerne une amélioration du flot de conception standard. La deuxième, l'approche unifiée, consiste à utiliser les langages de haut niveau – C/C++, SystemC, MATLAB® – en conception matérielle des systèmes complexes.

Mots clés : Détection à usagers multiples (*MUD : Multiuser detection*), communication WCDMA, architecture VLSI, méthodologie de conception, parallélisme, système-sur-une-puce (*SoC : System-on-chip*)

ABSTRACT

The interferences – intersymbol interferences (ISI) and multiaccess interferences (MAI) – are severely problematic for data reconstruction in the third generation (3G) wireless communication systems. In presence of MAI, the multiuser detection (MUD) becomes an essential part in deployment of 3G systems at receivers' level. The MUD has been making a continuous research topic since more twenty years. The algorithmic aspect of MUD is transparent. There exist several sophisticated algorithms permitting to effectively cancel MAI. Nevertheless, due to the implementation complexity while satisfying algorithmic performances, the aspect of VLSI implementation still remains non-trivial. So this thesis deals with VLSI implementation and design methodology of multiuser receivers.

The major contribution of this doctoral thesis consists in proposing and developing the parallel VLSI architectures of a class of adaptive MUD algorithms. These architectures satisfy the algorithmic performances for several data rates in several WCDMA communication scenarios. Also, the studied receiver was extended and implemented for data multirate systems. Considering limitations of commercial tools, and limitations of design methodologies actually, the second contribution of this work deals with the design methodologies. It consists of two straightforward design methodologies for complex systems using commercial tools. The first one, namely conventional approach, concerns an improvement in the standard design flow. The second one, namely unified approach, consists in using the high level languages – C/C++, SystemC and MATLAB® – in hardware design of complex systems.

Key words: Multiuser detection (MUD), WCDMA communication, VLSI architecture, design methodology, parallelism, system-on-a-chip (SoC).

REMERCIEMENTS

Les travaux de ma thèse ont été effectués au sein du Laboratoire des Signaux et Systèmes Intégrés (LSSI) à l'université du Québec à Trois-Rivières (UQTR) sous la direction de Monsieur Daniel Massicotte – Professeur titulaire à l'UQTR. Je remercie vivement mon directeur de m'avoir accueilli dans son laboratoire, d'avoir supervisé ma thèse d'une manière remarquable, et d'avoir continuellement financé mes recherches et mes études durant ces quatre dernières années. Je lui remercie pour sa direction, sa disponibilité, sa patience et sa confiance en ma liberté dans ma recherche. Qu'il trouve toute ma gratitude pour l'opportunité de découvrir les frontières de mes intérêts couvrant dans cette thèse qui me demandent une énergie permanente mais qui me sont stimulantes et intéressantes, et l'opportunité que j'ai entreprise dans plusieurs choses à côté de ma thèse doctorale. Enfin, je me trouve moi-même chanceux parmi ses élèves pour tous ceux qui m'ont été consacrés durant ma vie académique intéressante à Trois-Rivières.

Je voudrais remercier Monsieur Adel Omar Dahmane – Professeur à l'UQTR pour ses discussions constructives et utiles, notamment sur l'aspect algorithmique du récepteur CF-MUD. Je le remercie aussi d'avoir accepté d'évaluer ma thèse et participer au jury de ma soutenance.

Je tiens à remercier Monsieur El Mostapha Aboulhamid – Professeur titulaire à l'université de Montréal – pour son temps qui m'a dédié durant mon séjour d'un mois au sein de son équipe LASSO, et de ses commentaires constructifs et productifs. Je le remercie d'avoir accepté d'évaluer mon examen doctoral, et cette fois ma thèse, et de participer au jury à ma soutenance.

J'aimerais remercier Monsieur Sébastien Roy – Professeur à l'université Laval – d'avoir accepté d'évaluer ma thèse et de participer au jury de ma soutenance.

Je tiens notamment à remercier tous les évaluateurs pour leurs commentaires et suggestions qui me sont grandement utiles pour améliorer la clarté de la version finale de ma thèse.

J'aimerais remercier tous mes collègues au Laboratoire LSSI avec qui j'ai partagé l'environnement agréable de travail pendant ces dernières années, incluant en ordre alphabétique : Adel Omar Dahmane, Messaoud Ahmed Ouameur, Didier Félix, Élie Sarraf, François Nougrou, Hugues Paquin, Mohammed Elnamaky, Mohammed Zebdi, Patrick-Henry Eyoum et Simon Boucher. J'aimerais aussi remercier les collaborateurs scientifiques du Laboratoire LSSI : Messieurs Patrice Quinton et Tanguy Risset – Professeurs de l'IRISA en France pour leurs discussions productives.

J'aimerais remercier le personnel de la société Axiocom inc. et le personnel du département de génie électrique et génie informatique de l'UQTR pour leur support. Notamment, Madame Sylvie Laporte de Axiocom inc. est grandement appréciée pour sa correction du rapport de mon examen doctoral, Madame Sylvie Desilets et Monsieur Guy Boisclair de l'UQTR sont grandement appréciés pour leur soutien administratif et informatique respectivement.

Je remercie divers organismes qui ont partiellement financé mes études et mes recherches : Axiocom inc. pour son support financier et technique, CRSNG pour son support financier à mes recherches et à ma participation aux conférences, Fondation et le département de génie électrique et génie informatique de l'UQTR pour les bourses que j'ai obtenues à plusieurs reprises.

Je voudrais remercier mes amis vietnamiens à Grenoble dont leurs humours, leurs courages étaient exemplaires. Aussi je remercie mes amis « internationaux » à Trois-Rivières qui m'ont offert des activités intéressantes et inoubliables. Ceux dont je parle se reconnaîtront.

Je n'oublie pas de remercier mes professeurs vietnamiens de Ho Chi Minh – ville et mes professeurs français de Grenoble. Ils ont su me motiver vers les frontières des domaines de mes intérêts.

Dernièrement mais certainement pas le moins important, je veux exprimer ma gratitude la plus grande à mes parents et à tous les membres de notre famille au Vietnam. Leur support moral et leur confiance immenses en moi m'ont constamment encouragé, malgré certaines périodes difficiles, pendant mes longues études supérieures à l'extérieur de chez nous. Enfin, Merci ma Hue-Ngoc pour son accompagnement, notamment durant ces derniers temps.

DÉDICACE

À ma mère Thi-Nam Lê et mon père Cung Hô

TABLE DES MATIÈRES

Résumé.....	iii
Abstract	iv
Remerciements.....	v
Dédicace.....	vii
Table des matières.....	viii
Liste des figures	viii
Liste des tableaux.....	viii
Liste des acronymes.....	viii
 Chapitre I Introduction.....	 1
I.1. Problématique	3
I.2. Contexte de travail et objectifs	6
I.3. Méthodologie.....	7
I.4. Conclusions	10
 Chapitre II Détection à usagers multiples.....	 11
II.1. Concepts de base	11
II.1.1. Principe du CDMA	11
II.1.2. Interférences à accès multiples	14
II.1.3. Modèle de système DS-CDMA	16
II.2. Détection à usagers multiples	20
II.2.1. Description générale de la MUD	20
II.2.2. Filtre apparié	22
II.2.3. Récepteur <i>Rake</i> et ses variantes	23
II.2.4. Récepteur optimal	25

II.2.5.	Décorrélateur.....	26
II.2.6.	Récepteur MMSE.....	26
II.2.7.	Récepteur SIC et ses variantes.....	27
II.2.8.	Récepteur PIC et ses variantes.....	29
II.2.9.	Complexité des récepteurs à usagers multiples	32
II.3.	Détection à usagers multiples multi-débit	33
II.3.1.	Modèle de système DS-CDMA multi-débit	34
II.3.2.	Méthodes de détection à usagers multiples multi-débit.....	37
II.3.2.A.	Méthode LRD	37
II.3.2.B.	Méthode HRD	41
II.3.2.C.	Méthode successive.....	43
II.4.	Travaux d'implémentation des récepteurs à usagers multiples	44
II.5.	Librairies arithmétiques en virgule fixe.....	49
II.5.1.	Proposition de IEEE.....	50
II.5.2.	Toolbox de MATLAB®	51
II.5.3.	Librairies arithmétiques de SystemC	51
II.5.4.	Types de données de Mentor Graphics : Algorithmic C™.....	52
II.6.	Conclusions	53
Chapitre III	Architectures proposées du récepteur CF-MUD	54
III.1.	Description de l'algorithme CF-MUD	54
III.1.1.	Bloc Signature.....	56
III.1.2.	Bloc Détection	58
III.2.	Architecture générale.....	60
III.3.	Noyau du bloc Signature	63
III.4.	Noyau du bloc Détection	64
III.5.	Multiplexage temporel.....	65
III.6.	Architecture des mémoires	66
III.7.	Architecture de l'unité de traitement PE	71
III.8.	Parallélisme et stratégie de pipeline	78
III.9.	Résultats d'implémentation	81
III.10.	Conclusions	87
Chapitre IV	Architectures proposées du récepteur MUD multi-débit.....	88

IV.1. Extension de l'algorithme CF-MUD aux systèmes multi-débit	88
IV.1.1. Méthode LRD-Sign-MUD	90
IV.1.2. Méthode HRD-Sign-MUD.....	93
IV.2. Architecture matérielle des récepteurs multi-débit.....	97
IV.2.1. Architecture du récepteur LRD-Sign-MUD	97
IV.2.2. Architecture du récepteur HRD-Sign-MUD	98
IV.3. Résultats de simulations et d'implémentation	100
IV.4. Conclusions	108
Chapitre V Contribution à la méthodologie de conception.....	109
V.1. Flot de conception général.....	109
V.2. Approche conventionnelle	111
V.3. Approche unifiée	117
V.3.1. Motivations d'une approche unifiée	117
V.3.2. Approche unifiée basée sur SystemC	118
V.3.2.A. Cadre de cosimulation MATLAB®-SystemC	120
V.3.2.B. Expérimentations	122
V.3.3. Approche unifiée basée sur C/C++ standard	125
V.4. Développement des bibliothèques spécialisées en virgule fixe	127
V.4.1. Représentation des nombres en virgule fixe	127
V.4.2. Paquetage VHDL en virgule fixe.....	129
V.4.3. Classe de données en virgule fixe MATLAB® : AC_M.....	131
V.5. Conclusions	137
Chapitre VI Conclusions et perspectives.....	140
VI.1. Contribution d'implémentation d'algorithmes spécifiques	141
VI.2. Contribution méthodologique	142
VI.3. Application des résultats des travaux	143
VI.4. Perspectives de développements futurs	145
Publications.....	141
Bibliographie.....	143

Annexe A	Caractéristiques radio essentielles des systèmes WCDMA mode FDD	152
A.1.	Structure du canal physique.....	153
A.2.	Modulation et étalement de spectre	155
A.3.	Démodulation et désétalement de spectre.....	158
A.4.	Débit de données	158
A.5.	Quelques caractéristiques radio de mesure.....	159
Annexe B	Implémentation des séquences de codes de brouillage sans multiplication. 160	
B.1.	Générateur de séquences de brouillage	160
B.2.	Implémentation matérielle sans multiplication du générateur des séquences de codes de brouillage courts.....	162
Index		165

LISTE DES FIGURES

Figure 1 :	Problématique générale d'un système sans fil 3G.....	4
Figure 2 :	Principe de la communication CDMA.	12
Figure 3 :	Techniques du CDMA.....	13
Figure 4 :	Représentation des interférences à accès multiples.....	15
Figure 5 :	Modèle en bande de base d'un système DS-CDMA. Pour simplifier la synchronisation à la réception, on considère le même filtre impulsionnel ($\psi_1(t) = \psi_k(t) = \dots = \psi_K(t)$).	16
Figure 7 :	Classification des récepteurs à usagers multiples (liste non-exhaustive).	21
Figure 8 :	Principe du filtre apparié (MF).	22
Figure 9 :	Principe du récepteur Rake combinant L trajets de l'utilisateur k	23
Figure 10 :	Principe du décorrélateur (ZF).	26
Figure 11 :	Principe du récepteur MMSE avec combinaison des trajets multiples après.	27
Figure 12 :	Principe du récepteur MMSE avec combinaison des trajets multiples avant.	27
Figure 13 :	Principe du récepteur SIC à M étage : a) Structure bloc, b) Structure d'un étage m	28
Figure 14 :	Principe du récepteur MPIC à M étage au niveau des brides : a) Structure bloc, b) Structure d'un étage m	30
Figure 15 :	Principe du récepteur MPIC à M étage au niveau des bits : a) Structure bloc, b) Structure d'un étage m	31
Figure 16 :	Modèle en bande de base d'un système CDMA multi-débit, avec le même filtre impulsionnel ($\psi_1^1(t) = \dots = \psi_{K^1}^1(t) = \dots = \psi_1^G(t) = \dots = \psi_{K^G}^1(t) = \psi(t)$).	34
Figure 17 :	Représentation des symboles étalés multi-débit, par exemple le débit du groupe 1 est deux fois et quatre fois plus lent que le groupe 2 et le groupe G	36

Liste des figures

Figure 18 : Dérivation des usagers virtuels par la méthode LRD : a) Symboles de l'utilisateur k du groupe g , par exemple le débit du groupe g est quatre fois plus élevé que le groupe 1, b) Symboles virtuels dérivés.	39
Figure 19 : Principe de la méthode LRD basée sur un MUD mono-débit.	40
Figure 20 : Dérivation des usagers virtuels par la méthode HRD : a) Symboles de l'utilisateur k du groupe g , par exemple le débit du groupe g est quatre fois plus faible que le groupe G , b) Symboles virtuels dérivés.	42
Figure 21 : Principe de la méthode HRD basée sur un MUD mono-débit.	43
Figure 22 : Principe de la méthode multi-débit successive basée sur un MUD mono-débit.	44
Figure 23 : Principe du récepteur CF-MUD. La rétroaction des données est optionnelle.	55
Figure 24 : Principe de la cascade des filtres adaptatifs du bloc Signature.	56
Figure 25 : Principe de la cascade des filtres adaptatif du bloc Détection.	58
Figure 26 : Architecture matérielle générale du CF-MUD.	61
Figure 27 : Organisation des tableaux des PE, et des mémoires internes.	62
Figure 28 : Mapping du bloc Signature sur le tableau des PE et les mémoires internes. Les deux flèches parallèles signifient les données complexes.	64
Figure 29 : Mapping du bloc Détection sur le tableau des PE et des mémoires internes. Les deux flèches parallèles signifient les données complexes.	65
Figure 30 Mémoire dual-port.	68
Figure 31 : Structure partielle d'une mémoire semi-globale des poids adaptés ($\hat{\mathbf{w}}$ et $\hat{\mathbf{v}}$). <i>ADAPT</i> : phase d'adaptation, <i>DETECT</i> : phase de détection, Signature : bloc Signature, Détection : bloc Détection.	69
Figure 32 : Commutation d'accès mémoire de la phase d'adaptation et de la phase de détection. <i>ADAPT</i> : phase d'adaptation, <i>DETECT</i> : phase de détection. La flèche foncée signifie l'activation de l'accès mémoire, alors que la flèche pale signifie l'inactivation de l'accès mémoire.	70
Figure 33 : Structure du convertisseur Parallel2Serial : a) Diagramme bloc, b) Schéma RTL, c) Machine à états, d) Diagramme des états.	71
Figure 34 : Flot d'états du noyau FIR à données complexes.	74

Liste des figures

Figure 35 :	Flot d'états du noyau de l'adaptation LMS à données complexes avec erreur d'adaptation réelle.	75
Figure 36 :	Graphe du flot de données du processeur LMS à données complexes.....	76
Figure 37 :	Schéma RTL du processeur LMS à données complexes.....	77
Figure 38 :	Stratégie du pipeline de la phase d'adaptation.	79
Figure 39 :	Stratégie du pipeline de la phase de détection.....	80
Figure 40:	Principe du LRD-Sign-MUD basé sur la Signature adaptative mono-débit. Le récepteur fonctionne au niveau de brides avec période des symboles N^1	91
Figure 41 :	Principe du HRD-Sign-MUD basé sur la Signature adaptative mono-débit. Le récepteur fonctionne au niveau de brides avec période des symboles N_c^G	94
Figure 42 :	Architecture du récepteur LRD-Sign-MUD basée sur la Signature adaptative mono-débit.	98
Figure 43 :	Architecture du récepteur HRD-Sign-MUD basée sur la Signature adaptative mono-débit.	99
Figure 44 :	Circuit de MRC avec les additionneurs à deux entrées : a) Structure parallèle, b) Structure parallèle et pipeline c) Structure série et d) Structure série et pipeline...	100
Figure 45 :	Performances de la Signature adaptative par rapport au Rake et soft MPIC à 4-étage dans un système mono-débit de 16 usagers et de 64 kb/s.....	101
Figure 46 :	Performances du schéma LRD-Sign-MUD vs HRD-Sign-MUD avec un système à double débit ($G=2$) : 4 usagers à 64 kb/s et 4 usagers à 144 kb/s.	102
Figure 47 :	Performances virgule fixe du schéma LRD-Sign-MUD pour un système à double débit ($G=2$) : 4 usagers à 64 kb/s et 4 usagers à 144 kb/s.	103
Figure 48 :	Performances virgule fixe du schéma HRD-Sign-MUD pour un système à double débit : 4 usagers à 64 kb/s et 4 usagers à 144 kb/s.	104
Figure 49 :	Complexité matérielle du schéma LRD-Sign-MUD vs. HRD-Sign-MUD dans un système à double débit ($G=2$) en termes de nombre : a) d'additions, b) de multiplications, et c) d'accès mémoire.	106
Figure 50 :	Flot de conception général cible sur FPGA.....	111
Figure 51 :	Approche conventionnelle d'un flot de conception.....	113

Liste des figures

Figure 52 : Approche unifiée basée sur SystemC d'un flot de conception.....	119
Figure 53 : A cadre de cosimulation utilisant MATLAB® et SystemC.	121
Figure 54 : Cosimulation d'un récepteur CF-MUD utilisant MATLAB® et SystemC.....	123
Figure 55 : Données transmises de la figure 54 : a) Image originale, b) 8 sous-images bruitées après le canal de transmission, c) 8 sous-images reconstruites après le récepteur en virgule fixe.....	124
Figure 56 : Approche unifiée basée sur C/C++ standard d'un flot de conception.....	126
Figure 57 : Principe de la classe AC_M intégré sous MATLAB®.	132
Figure 58 : Structure d'une trame de la liaison montante.	154
Figure 59 : Étalement et modulation de la liaison montante : a) Étalement, b) Modulation. ...	156
Figure 60: Arbre des codes OVSF.	156
Figure 61 : Générateur des séquences des codes de brouillage courts (structure Fibonacci). ..	157
Figure 62 : Démodulation et désétalement de spectre basé sur un filtre apparié.....	158
Figure 63 : Structure Galois du générateur des codes de brouillage courts.	162
Figure 64 : Implémentation du modulo-4 dans la structure du générateur des codes de brouillage courts.	163
Figure 65 : Génération de la séquence $C_{short,n}(i)$	164

LISTE DES TABLEAUX

Tableau 1 : Méthodologie du projet doctoral en cinq étapes.....	8
Tableau 2 : Complexité des récepteurs à usagers multiples.	32
Tableau 3 : Résumé des travaux d'implémentation des récepteurs à usagers multiples.	49
Tableau 4 : Paramètres de reconfiguration de l'architecture	63
Tableau 5 : Nombre de cycles d'horloge nécessaires d'un PE.....	78
Tableau 6 : Nombre maximum d'usagers (\hat{K}) qui peut être intégré sur un seul dé de plusieurs composants de la famille Virtex-II Pro.....	82
Tableau 7 : Nombre maximum d'usagers (\hat{K}) qui peut être intégré sur un seul dé de plusieurs composants de la famille Virtex-II.	82
Tableau 8 : Taux d'utilisation de ressources matérielles sur un seul dé de plusieurs composants de la famille Virtex-II Pro (correspond avec le tableau 6).	83
Tableau 9 : Taux d'utilisation de ressources matérielles sur un seul dé de plusieurs composants de la famille Virtex-II (correspond avec le tableau 7).....	83
Tableau 10 : Nombre de secteurs capable être traité sur un seul composant de la famille Virtex-II Pro.....	84
Tableau 11 : Taux d'utilisation de ressources matérielles pour le nombre de secteurs dans une même BTS du tableau 10 dans les composants de la famille Virtex-II Pro.	84
Tableau 12 : Résultats après placement & routage à l'aide de l'outil physique de Xilinx ciblé sur le composant Virtex-II Pro XC2VP30 pour un système avec $K = 16$ usagers dans l'environnement d'intérieur (stationnaire) et piétons (semi-stationnaire).	85
Tableau 13 : Comparaison des travaux proposés par rapport aux travaux d'implémentation du Tableau 3.	87
Tableau 14 : Complexité matérielle des schémas multi-débit de Sign-MUD.	105

Tableau 15: Nombre maximum d'utilisateurs équivalent (K^{344V}) des systèmes multi-débit sur les composants de la famille Virtex II Pro pour la méthode LRD-Sign-MUD ou HRD-Sign-MUD.	107
Tableau 16 : Nombre d'utilisateurs implémenté sur les composants de la famille Virtex II Pro et le taux d'utilisation respectif du schéma LRD du récepteur CF-MUD multi-débit...108	
Tableau 17 : Nombre d'utilisateurs implémenté sur les composants de la famille Virtex II Pro et le taux d'utilisation respectif du schéma HRD du récepteur CF-MUD multi-débit...108	
Tableau 18 : Conditions de simulations du bloc Signature d'un récepteur CF-MUD.	123
Tableau 19 : Temps de simulation du bloc Signature en virgule flottante et fixe.....	125
Tableau 20 : Comparaison du paquetage des arithmétiques à virgule fixe <i>lssi_std_logic_fixed</i> par rapport aux autres propositions.....	130
Tableau 21 : Opérateurs arithmétiques de surcharge de la classe AC_M.	133
Tableau 22 : Opérateurs logiques.	134
Tableau 23 : Opérateurs de conversion.	134
Tableau 24 : Fonctions C/C++ utilisables en environnement MATLAB®.....	135
Tableau 25 : Benchmark avec un égaliseur FIR-LMS et un récepteur CF-MUD de la classe AC_M proposée par rapport à l'objet <i>fi</i> de MATLAB®.....	135
Tableau 26 : Benchmark avec opérations arithmétiques de la classe AC_M proposée par rapport à l'objet <i>fi</i> de MATLAB®.....	136
Tableau 27 : Benchmark avec opérations arithmétiques utilisant les DLLs de la classe AC_M proposée par rapport à l'objet <i>fi</i> de MATLAB®.....	137
Tableau 28 : Paramètres de conception importants de la norme WCDMA.	153
Tableau 29 : Débits de données des facteurs OVVSF types.	159
Tableau 30 : Conditions de propagation pour des environnements d'évanouissement multi-trajet.159	
Tableau 31 : Mapping de la séquence $z_n(i)$ aux séquences $c_{short,1,n}(i)$ et $c_{short,2,n}(i)$	162

LISTE DES ACRONYMES

ALAP	As late as possible
ALU	Arithmetic and logic unit/ Unité arithmétique et logique (UAL)
AM	Amplitude modulation
AMPS	Advanced mobile phone service
AMRC	Accès multiple à répartition en code/ Code division multiaccès (CDMA)
ASAP	As soon as possible
ASIC	Application specific integrated circuit
ASSP	Application specific standard processor
BER	Bit error rate
BPSK	Binary phase shift keying
BRAM	Block RAM (sur les composants FPGA Virtex-II et Virtex-II Pro de Xilinx®)
BSC	Binary symmetric channel
BTS	Base transceiver station
CDMA	Code division multiple access/ Accès multiple à répartition de codes (AMRC)
CF-MUD	Cascade filter multiuser detector
CRSNG	Conseil de recherches en sciences et en génie du Canada
DFG	Data flow graph
DS-CDMA	Direct-sequence CDMA
DSP	Digital signal processing/processor
FDD	Frequency division duplex
FH-CDMA	Frequency hopping CDMA
FIR	Finite impulse response (filter)
FM	Frequency modulation

Liste des acronymes

FPGA	Field programmable gate array
GPS	Global positioning system / Système de positionnement mondial
GSM	Global system for mobile communications
HDL	Hardware description language
HRD	High rate detector (scheme)
ISI	Intersymbol interference
LMS	Least mean square (filter)
LRD	Low rate detector (scheme)
LSSI	Laboratoire des signaux et systèmes intégrés/ Laboratory of signal and system integration
MAI	Multiaccess interference / Interférence à accès multiples
MAP	Maximum a posteriori probability
MLSE	Maximum likelihood sequence estimation
MMSE	Minimum mean square error
MRC	Maximum ratio combining
MUD	Multiuser detector/ Multiuser detection
OSCI	Open SystemC Initiative
OVSF	Orthogonal variable spreading factor
PE	Processing element
PIC	Parallel interference canceller
PM	Phase modulation
PN	Pseudo noise
QPSK	Quadrature phase shift keying
RAM	Random access memory
RLS	Recursive least square
RTL	Register transfer level
SIC	Serial interference canceller
SoC	System-on-a-chip

Liste des acronymes

SOPC	System-on-a-programmable-chip
SSMA	Spreading spectrum multi-access
TDD	Time division duplex
TD-SCDMA	Time division - synchronous CDMA
TH-CDMA	Time Hopping CDMA
UAL	Unité arithmétique et logique / Arithmetic and logic unit (ALU)
UQTR	Université du Québec à Trois-Rivières
UTRAN	UMTS terrestrial radio access network
VHDL	Very high speed integrated Hardware Description Language
VLIW	Very long instruction word
VLSI	Very large scale integration
WCDMA	Wideband CDMA

CHAPITRE I

INTRODUCTION

La communication sans fil de troisième génération (3G) a été adoptée pour les besoins de hauts débits et d'utilisation efficace du canal spectral. Les débits escomptés de la 3G sont 144 kb/s pour les véhicules, 384 kb/s pour piétons et 2 Mb/s pour environnements d'intérieur [87], [117]. De plus, elle permet l'itinérance globale (mondialement). Parmi plusieurs propositions de standardisation de la 3G à travers le monde [117], deux normes les plus importantes ont été retenues : WCDMA, et cdma2000. La différence majeure entre ces normes consiste en la largeur de bande passante, la fréquence des brides (*chip frequency*), la structure du canal descendant, et les mécanismes de synchronisation au niveau de station de base (*BTS : Base transceiver station*) [58], [87]. Cependant, ces techniques de la 3G sont communément basées sur l'accès multiple à répartition de codes à séquences directes (*DS-CDMA : Direct sequence - CDMA*). D'ailleurs, un autre standard important a été proposé en Chine : TD-SCDMA (*Time Division – Synchronous CDMA*). En fait, la Chine a adopté cette norme, qui a été conjointement développée par Siemens AG et l'académie chinoise de la technologie des télécommunications, en vue d'être indépendant dans ses développements et déploiements des systèmes 3G dans le marché local [101]. La norme TD-SCDMA combine CDMA en mode TDD et mode synchrone. Le mode TDD de TD-SCDMA permet une utilisation asymétrique du spectre de la liaison montante et descendante. La norme TD-SCDMA impose que les signaux des usagers soient synchronisés aux BTS comme dans les systèmes CDMA synchrones. En plus, la norme TD-SCDMA offre certains avantages aussi bien technologiques qu'économiques [101].

Dans un système CDMA, les usagers actifs simultanément ont accès à la bande fréquentielle en même temps mais avec les signatures (ou codes d'étalement¹), distinctes. Ces signatures consistent en les codes orthogonaux assignés à chaque usager permettant de détecter les données transmises [110]. À la réception, en connaissant le code de l'utilisateur d'intérêt, le récepteur peut décoder le signal reçu et extraire l'information transmise en utilisant un filtre apparié (*MF : Matched filter*) [cf. II.2.2] à cette signature [110]. Le récepteur *Rake* [cf. II.2.3] a été proposé et utilisé dans les systèmes CDMA IS-95A de la deuxième génération (*2G : 2nd generation*) [87], [110]. Le *Rake*, consistant à combiner plusieurs trajets – normalement 3 à 6 – d'un même usager dont chaque trajet est détecté par un filtre MF apparié à la signature de cet usager, améliore considérablement les performances par rapport aux filtres MF.

Les performances du récepteur *Rake* sont garanties à condition que les codes d'étalement soient parfaitement orthogonaux. Toutefois, l'orthogonalité parfaite des codes est difficile à atteindre dans la conception de ceux-ci d'une part [32], [44], [61], et d'autre part, l'orthogonalité est encore diminuée par l'évanouissement (*fading*) du canal de transmission [109]. Ainsi, l'application d'un récepteur *Rake* seul n'est pas suffisante pour annuler des interférences à accès multiples (*MAI : Multiple access interference*) [cf. II.1.2], car le *Rake* ne tient pas compte de MAI. Les récepteurs de la 3G doivent tenir compte non seulement des interférences inter-symboles (*ISI : Intersymbol interference*), mais aussi, et plus important, des MAI. La détection à usagers multiples (*MUD : Multiuser detection*)² devient alors une partie essentielle dans le déploiement des systèmes 3G au niveau des récepteurs [7], [20], [30], [79], [109]. Toutefois, l'insertion d'une fonction de MUD aux infrastructures des systèmes existants qui utilisent des récepteurs *Rake* est souhaitée par les opérateurs réseaux mais très coûteuse pour les fabricants.

La détection à usagers multiples fait l'objet de recherches continues depuis plus de vingt ans maintenant à compter de la proposition du récepteur optimum de Verdú [109]. Les domaines

¹ Certains auteurs utilisent le terme « code signature », certains d'autres utilisent « code d'étalement ». Dans ce manuscrit, « code signature » et « code d'étalement » sont indifféremment utilisés pour désigner un code distingué de chaque usager dans un système DS-CDMA.

² Dans ce manuscrit, l'acronyme MUD est aussi utilisé pour désigner le récepteur à usagers multiples.

de recherche relatifs à celle-ci sont très vastes. On peut citer quelques domaines de recherche relatifs ci-dessous.

- L'étude sur l'interface terrestre pour la 3G. L'objectif consiste à étudier et standardiser l'interface radio (bande passante, fréquence des brides, etc.) pour les systèmes de communications 3G [82].
- La conception des codes orthogonaux. L'objectif consiste à étudier et concevoir des codes signatures (codes d'étalement) dont l'orthogonalité est maximisée [32], [61].
- L'étude des antennes intelligentes. Une antenne intelligente consiste en un réseau d'antennes, combiné avec le traitement des signaux dans le temps et dans l'espace. On reconnaît les deux types d'antennes qui sont applicables aux systèmes de communications : *Beamforming* numérique et antenne adaptative [27].

Dans le cadre de notre travail, nous nous intéressons aux techniques de MUD, notamment l'aspect de l'implémentation VLSI en temps réel. Spécifiquement, dans ce travail de thèse, nous nous intéressons plus particulièrement à une classe de récepteurs adaptatifs, nommée CF-MUD. L'algorithme CF-MUD est développé à l'Université du Québec à Trois-Rivières au sein du Laboratoire des signaux et systèmes intégrés en collaboration avec la société Axiocom inc.

I.1. Problématique

Dans un canal de communication sans fil terrestre, le signal transmis d'un usager d'intérêt subit plusieurs atténuations comme présentées à la figure 1.

Premièrement, il se propage à travers plusieurs trajets, à cause des réflexions et/ou diffractions étant données des obstacles comme des bâtiments, des montagnes, etc. Cela cause ainsi des interférences ISI dues aux trajets multiples. Les interférences ISI consistent en plusieurs copies avec délais et amplitudes différents du signal d'un même usager d'intérêt au récepteur. Chaque copie du signal est alors caractérisée par un profil puissance-délai (amplitude et délai).

Pour l'utilisateur 1 (MS_1) dans la figure 1, on considère pour des raisons de simplification 2 trajets. Le profil puissance-délai de chaque trajet de l'utilisateur 1 est respectivement caractérisé par $A_{1,1}, \tau_{1,1}$ et $A_{1,2}, \tau_{1,2}$.

Dans un canal à usagers multiples basé sur le CDMA, le signal reçu d'un usager d'intérêt est toujours superposé aux signaux des autres usagers existant simultanément dans le système. Les signaux des autres usagers superposés au signal de l'utilisateur d'intérêt constituent ainsi les interférences MAI [cf. II.1.2]. Dans la figure 1, le signal reçu de l'utilisateur 1, qui est l'utilisateur d'intérêt par exemple, à la réception est superposé par le signal de l'utilisateur 2 et les signaux de tous les autres usagers existants. La méthode simple et largement utilisée dans les systèmes actuels basée sur le récepteur *Rake*, consiste à traiter les signaux de tous les autres usagers actifs comme un bruit blanc additif qui se superpose au signal de l'utilisateur d'intérêt.

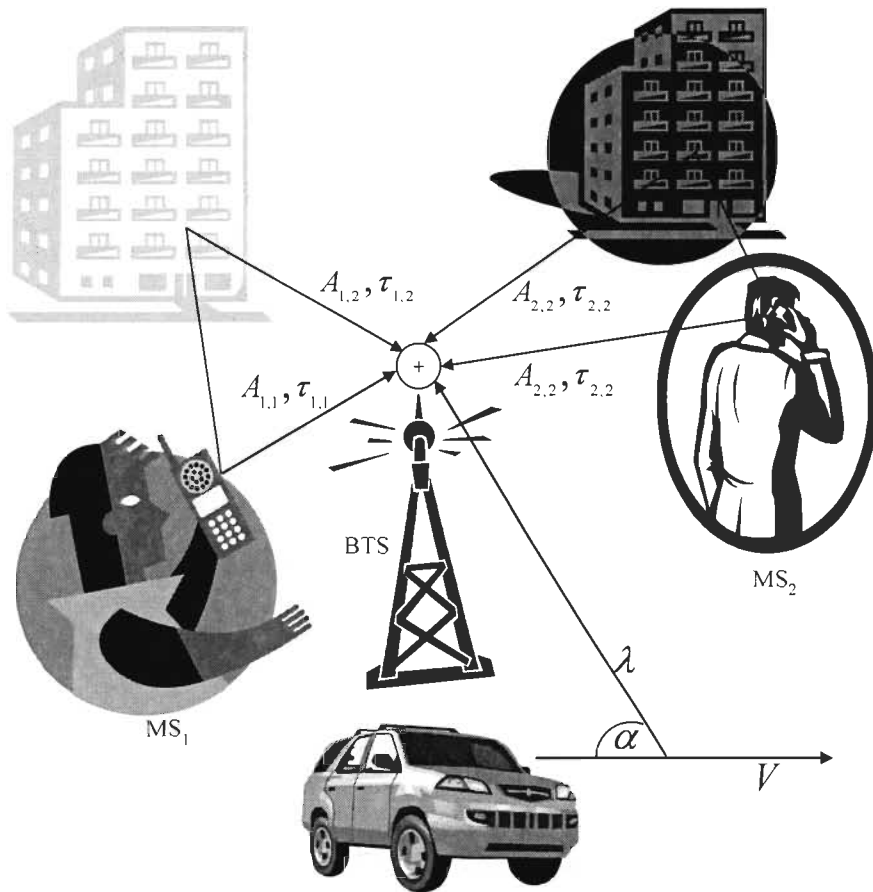


Figure 1 : Problématique générale d'un système sans fil 3G.

Due à l'atténuation des puissances (amplitudes) pendant la transmission dans un canal, les signaux reçus au récepteur subissent le problème proche-loin (*near-far*) [58]. C'est-à-dire que plus le signal transmis est proche du récepteur, plus sa puissance (amplitude) reçue est importante et vice-versa. En considérant l'utilisateur 1, à la figure 1, le trajet 1 (direct) de son signal est plus court que le trajet 2 (reflet sur un bâtiment). L'amplitude (puissance) relative au trajet 1 est donc plus importante que celle de son trajet 2. Ce phénomène complique la procédure d'annulation des interférences MAI étant donné que les puissances des signaux reçus ne sont pas les mêmes. En fait, la méthode commune, qui est utilisée dans certains récepteurs standards en présence du problème proche-loin – *Rake*, récepteur série (*SIC : Serial interference canceller*) –, consiste à reconnaître d'abord, par une méthode d'identification des canaux, les amplitudes des usagers pour augmenter les performances. Ensuite, ces récepteurs détectent successivement les données des usagers en ordre décroissant de leurs puissances respectives. Par conséquent, les performances de ces récepteurs – *Rake* et *SIC* – sont sensibles aux estimateurs des puissances [79], [109]. Ainsi, la présence du phénomène proche-loin exige que les paramètres des canaux de transmission – amplitudes et délais – soient exactement estimés. Une erreur d'estimation des canaux perturbe davantage les performances de ces derniers [79], [109].

Quand un transmetteur se déplace à une certaine vitesse, il subit l'effet Doppler comme présenté à la figure 1, lequel constitue un évanouissement sélectif en temps [58]. Un transmetteur, dont la longueur d'onde de sa porteuse est λ , et qui se déplace à une vitesse V avec un angle α par rapport au récepteur, subit un glissement fréquentiel de f_d calculé comme ci-dessous.

$$f_d = \frac{V}{\lambda} \cos(\alpha) \quad (1)$$

Quand les conditions de propagation changent en temps, normalement dues à la mobilité des transmetteurs, les paramètres canal changent. Si cela n'affecte que les amplitudes mais n'introduit pas de distorsion de la forme d'onde de la signature, on parle alors d'évanouissement à fréquence plate. Il existe plusieurs modèles pour caractériser le changement des amplitudes reçues, par exemple Rayleigh, Rice, Nakagami, etc. [109].

Dans plusieurs scénarios de communication à accès multiples, le changement en temps survient non seulement aux amplitudes des signaux reçus, mais aussi aux formes d'onde des signatures. Le changement des formes d'onde en temps est dû à la distorsion du canal. Dans ce cas de changement des formes d'onde des signatures, on parle alors d'évanouissement sélectif en fréquence.

Ainsi, dans les systèmes 3G à base du CDMA, les interférences constituent un problème majeur pour la reconstruction des données à la réception. En effet, il existe plusieurs algorithmes sophistiqués de la MUD. Toutefois, les travaux d'implémentation VLSI des algorithmes complexes restent encore non-triviaux. Cette limitation attire grandement notre attention dans le cadre de cette thèse.

I.2. Contexte de travail et objectifs

À l'heure actuelle, les systèmes de la 3G comptent un nombre d'abonnés à croissance exponentielle. Les systèmes 3G supportent plusieurs fonctions avancées qui sont imposées par les normes. Toutefois, la détection à usagers multiples est exclue par les fabricants bien qu'elle soit recommandée dans la norme 3GPP (cf. tableau 28 de l'annexe A). La détection à usagers multiples implique un grand changement dans la gestion des systèmes de communication CDMA d'une part [108]. D'autre part, elle impose des défis aussi bien à la recherche qu'à l'industrie. Le laboratoire LSSI à l'UQTR se met alors l'accent sur les travaux de recherche sur la détection à usagers multiples en étroite collaboration avec l'industrie. Le laboratoire a développé, entre autres, une classe d'algorithmes d'annulation des interférences à usagers multiples pour les BTS. Cette classe d'algorithmes de MUD est favorable pour une implémentation matérielle. L'implémentation matérielle d'une telle classe d'algorithmes permet alors de renforcer les performances des systèmes 3G face au problème MAI.

L'objectif principal de ce travail de recherche vise à proposer et à développer les architectures VLSI parallèles d'une classe d'algorithmes adaptatifs dédiés à l'annulation des interférences à accès multiples des récepteurs WCDMA développée au Laboratoire LSSI. Les

architectures développées doivent maximiser le nombre d'utilisateurs sur le même composant cible des FPGA tout en satisfaisant en même temps les contraintes aussi bien algorithmiques que d'implémentation matérielle. Les contraintes consistent à satisfaire les contraintes d'exécution en temps réel, les performances en termes de taux d'erreur BER, la précision arithmétique en virgule fixe, et bien sûr les contraintes associées au WCDMA. Cet objectif nous a aussi mené à étendre et implémenter ce récepteur pour des systèmes de communications multi-débit de données. L'atteinte de ces objectifs nous a finalement mené à proposer une méthodologie de conception sous contraintes matérielles et temporelles des architectures parallèles. Il s'agit d'une approche conventionnelle et d'une approche unifiée.

Les sous-objectifs de ce travail doctoral consistent ainsi à :

- Proposer et développer les architectures VLSI d'une classe d'algorithmes d'annulation, nommée CF-MUD (*Cascade Filter Multiuser Detector*), ciblée sur FPGA.
- Étendre et implémenter l'algorithme CF-MUD existant pour les systèmes multi-débit de données.
- Proposer une méthodologie de conception clairement établie des systèmes complexes en utilisant les outils commerciaux. Ainsi, deux approches ont été élaborées : l'approche conventionnelle et l'approche unifiée. L'approche conventionnelle porte sur une amélioration dans le flot de conception standard. L'approche unifiée consiste à utiliser les langages de haut niveau – C/C++, SystemC, MATLAB® – en conception matérielle des systèmes complexes.

I.3. Méthodologie

La méthodologie de recherche peut être résumée en cinq étapes comme présentées au tableau 1.

- Étape 1 : La recherche bibliographique vise à étudier les bases sur les récepteurs des systèmes CDMA dans le contexte des normes 3G, notamment WCDMA. L'étude

s'effectue d'abord sur l'aspect algorithmique puis sur l'aspect d'implémentation VLSI en temps réel. Dans cette étape, l'étude comparative des architectures VLSI des MUD proposées dans la littérature permet de choisir des travaux les plus pertinents pour comparer les performances des méthodes développées au LSSI dont le CF-MUD pour des systèmes WCDMA. Ainsi, nous avons pu établir l'ensemble des contraintes pour notre architecture en vue d'intégrer l'algorithme CF-MUD.

- Étape 2 : Nous avons établi et proposé les architectures ciblées sur FPGA du CF-MUD mono-débit. Ces travaux de l'étape 2 ont été possibles grâce à une compréhension de l'algorithme. Pour ce faire, nous avons réalisé une plateforme MATLAB® et C/C++ pour simuler dans l'environnement MATLAB®. Cette plateforme a également servi à des fins de simulations matérielles de nos propositions architecturales. Les travaux majeurs de cette étape consistent à concevoir et à développer les architectures proposées.

Tableau 1 : Méthodologie du projet doctoral en cinq étapes.

Étape 1	Recherche bibliographique et étude comparative des travaux pertinents dans la littérature
Étape 2	Proposition d'architectures d'un MUD pour les systèmes WCDMA sur FPGA
Étape 3	Développement d'une méthodologie d'implémentation en VLSI d'algorithmes avec contraintes
Étape 4	Étude de performances et amélioration pour multi-débit WCDMA
Étape 5	Extension de l'algorithme CF-MUD pour les systèmes multi-débit et son implémentation VLSI

- Étape 3 : Nous avons développé une méthodologie d'implémentation systématique sous contraintes. Il s'agit des contraintes des applications (dans notre cas de MUD, ce sont les contraintes de communication WCDMA en temps réel et les contraintes de performances algorithmiques attendues de MUD) et contraintes de ressources limitées des cibles comme les FPGA.

- Étape 4 : Nous avons étudié les performances des architectures que nous avons développées. Cette étude de performances permet l'amélioration du développement matériel en tenant compte de toutes les limitations de l'architecture développée dans l'étape 2. Également, les travaux de cette étape nous ont aidés dans le développement de l'architecture pour des systèmes multi-débit.
- Étape 5 : L'algorithme CF-MUD est étendu pour les systèmes multi-débit. L'algorithme étendu résultant est connu alors comme MUD multi-débit. L'implémentation de l'algorithme MUD multi-débit a été aussi effectuée. L'implémentation de ce récepteur résultant a été basée sur le noyau de CF-MUD mono-débit.

Le reste de ce manuscrit se structure de la manière suivante :

Le chapitre II – « Détection à usagers multiples » – fait office d'une synthèse de la détection à usagers multiples et des méthodes multi-débit dans la littérature. Il résume notamment l'état de l'art concernant l'implémentation VLSI des récepteurs à usagers multiples. Aussi, il résume succinctement la base de la communication CDMA.

Le chapitre III – « Architectures proposées du CF-MUD » – présente les architectures VLSI d'une classe de récepteurs à usagers multiples adaptatifs mono-débit. Il s'agit d'une architecture régulière et modulaire ciblée sur FPGA.

Le chapitre IV – « Architectures proposées du MUD multi-débit » – vise à étendre et implémenter le récepteur CF-MUD pour les systèmes multi-débit. Les deux méthodes multi-débit couramment utilisées dans la littérature ont été élaborées et adoptées pour étendre l'algorithme CF-MUD dans les contextes WCDMA multi-débit.

Le chapitre V – « Contribution à la méthodologie de conception » – présente nos contributions à la méthodologie de conception. Il s'agit de deux approches : approche conventionnelle et approche unifiée.

Le chapitre VI – « Conclusions et perspectives » – est consacré à synthétiser les résultats de la thèse, et tirer les conclusions et perspectives de ce travail doctoral.

L'annexe A – « Caractéristiques des systèmes WCDMA mode FDD » résume les caractéristiques radio essentielles de la norme WCDMA. Ces caractéristiques sont nécessaires pour une compréhension de techniques MUD et des travaux de cette thèse.

L'annexe B – « Implémentation des séquences de codes de brouillage sans multiplication » présente une implémentation efficace des codes de brouillage (*scrambling*) courts sans multiplication. Ces codes sont utilisés pour réaliser l'opération de bouillage dans les systèmes WCDMA.

I.4. Conclusions

La technologie des communications sans fil évolue rapidement, et nous vivons actuellement sa troisième génération. La 3G offre plusieurs avantages aussi bien au niveau technique qu'au niveau politique. Elle connaît déjà le nombre d'abonnés croissant à travers le monde. Toutefois, il existe plusieurs problèmes dans un système de communications sans fil 3G, plus particulièrement à la couche physique tels que le problème proche-loin, les interférences ISI et MAI et l'effet Doppler. Dans le cadre de ce travail de thèse, on s'intéresse plus particulièrement à un des problèmes les plus importants au niveau des récepteurs et à la couche physique : l'implémentation d'un MUD. En fait, les travaux relatifs à la détection à usagers multiples sont énormes et bien documentés au niveau algorithmique. Toutefois, les travaux d'implémentation VLSI restent encore non triviaux. Cette limitation attire ainsi notre attention dans le cadre de ce travail de recherche.

CHAPITRE II

DÉTECTION À USAGERS MULTIPLES

II.1. Concepts de base

II.1.1. Principe du CDMA

Le principe de la communication CDMA est présenté à la figure 2. Dans un tel système, chaque usager est assigné à une séquence de code d'étalement unique qui est utilisée pour étaler, ou coder, le signal portant l'information.

En connaissant ce code, le récepteur peut désétaler (décoder) et extraire l'information de cet usager. Ceci est possible car l'inter-corrélation entre les codes signatures des usagers est faible. Comme la bande passante du code d'étalement est choisie beaucoup plus large que celle du signal contenant l'information, le spectre de l'information est alors élargi ou étalé par le spectre du code. Ainsi, le CDMA est également appelé l'accès multiples à étalement de spectre (*SSMA : Spreading spectrum multiaccess*). De la même manière, le signal résultant est appelé signal à étalement de spectre. Dans les systèmes WCDMA, la fréquence des brides est à 3.84 MHz.

Tout système CDMA doit satisfaire les deux critères suivants :

- La bande passante du code d'étalement B_c doit être beaucoup plus large que celle de l'information B_i . Le rapport $N_c = B_c/B_i$ est donc, par définition, le taux d'étalement.

- La bande passante résultante ne dépend pas de l'information. Ceci est différent par rapport aux techniques classiques de modulation comme la modulation en fréquence (*FM : Frequency modulation*) et modulation en phase (*PM : Phase modulation*).

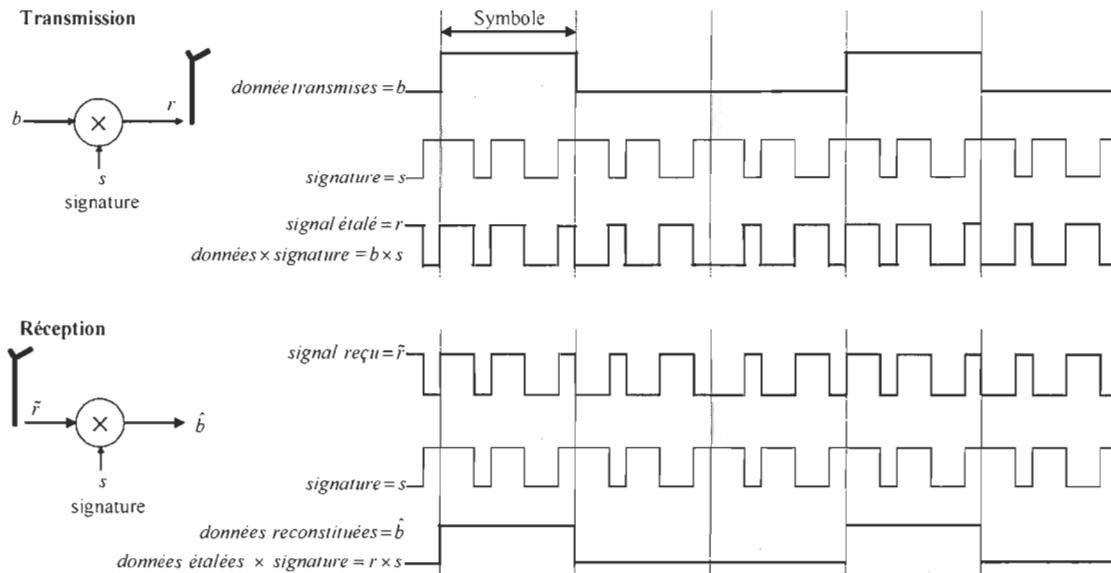


Figure 2 : Principe de la communication CDMA.

Il existe plusieurs techniques de modulation qui génèrent les signaux étalés en spectre dans la communication CDMA. Ces techniques de modulation différencient aussi les systèmes CDMA. Or, dans le cadre ce travail, nous nous intéressons notamment à la technique dans laquelle chaque usager est assigné à une séquence de code d'étalement distincte. Nous l'appelons alors la technique accès multiple par répartition de code à séquence directe (*DS-CDMA : Direct sequence CDMA*). D'autres techniques existent également : CDMA à saut fréquentiel (*FH : CDMA - Frequency hopping CDMA*), CDMA à saut temporel (*TH-CDMA : Time hopping CDMA*) et CDMA hybride. Dans la technique FH-CDMA, la fréquence du signal portant l'information varie (rapidement ou lentement) selon le code. Dans la technique TH-CDMA, le signal portant l'information n'est pas continuellement transmis mais par groupes de données en rafales courtes (*short bursts*) dont la position temporelle est déterminée par le code. La technique hybride combine deux ou plus de ces techniques de modulation afin d'exploiter les avantages combinés de celles-ci. La figure 3 résume la classification de ces techniques CDMA [87].

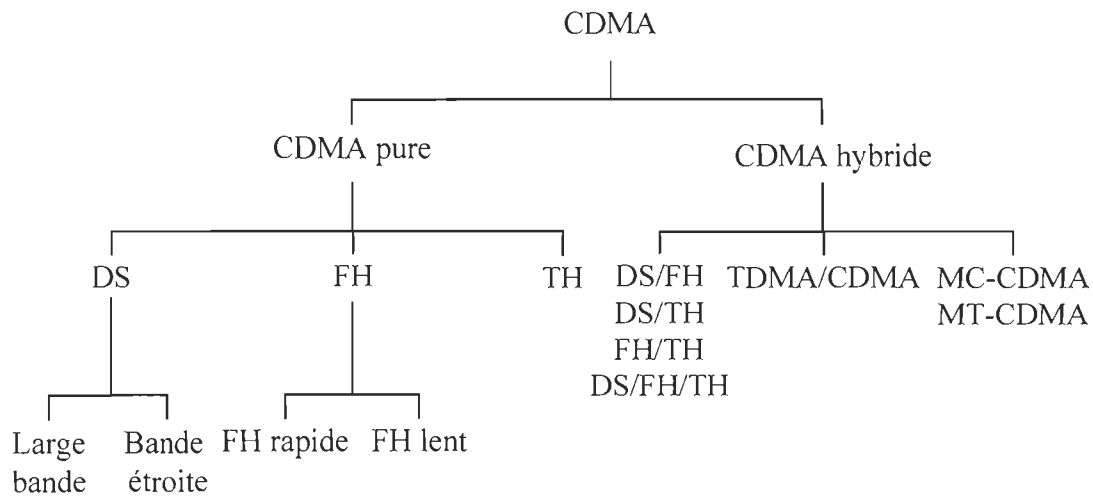


Figure 3 : Techniques du CDMA.

Les principales propriétés de la communication sont les suivants :

- Possibilité d'accès multiple : Tous les usagers instantanés bénéficient de la même bande passante à tout instant. Le récepteur les distingue grâce aux codes d'étalement, dont l'inter-corrélation est faible. Corréler le signal étalé d'un usager d'intérêt avec son code d'étalement permet de désétaler ce signal alors que les signaux des autres usagers restent en large bande.
- Protection des interférences multi-trajets : Dans le domaine fréquentiel, l'addition des signaux d'interférences au récepteur est constructive ou destructive à certaines fréquences. Dans le domaine temporel, ceci résulte en une dispersion du signal. La modulation à étalement de spectre peut combattre ce problème multi-trajet. Toutefois, la façon à laquelle le rejet des interférences multi-trajets s'effectue dépend fortement du type de modulation.
- Intimité : Le signal à étalement de spectre transmis est désétalé et l'information est extraite si et seulement si le récepteur connaît son code d'étalement.
- Rejet d'interférences : Inter-corréler le signal du code d'étalement avec le signal de bande étroite étale la puissance du signal de bande étroite, donc réduit la puissance des

interférences dans la bande passante de l'information. Ainsi, une certaine robustesse est obtenue contre le brouillage hostile, notamment brouillage à bande étroite.

- Probabilité d'interception faible : Le signal étalé est difficile à détecter car sa puissance est faible. La probabilité d'interception faible et la robustesse contre le brouillage hostile rendent modulation à étalement de spectre attrayante pour les applications militaires.

Les cinq propriétés ci-dessus sont communes dans les systèmes CDMA. De plus, le WCDMA supporte plusieurs vitesses de transmission de données. Ainsi, le concept de largeur de bande à la demande est bien implémenté. Chaque usager alloue une trame de 10ms dans laquelle la vitesse est constante. Les BS fonctionnent de manière asynchrone. Ainsi, elles n'ont plus besoin de référence temporelle comme dans les systèmes de positionnement mondial (*GPS : Global positioning system*). Les caractéristiques radio essentielles du WCDMA sont résumées dans la section II.4.

II.1.2. Interférences à accès multiples

Les interférences sont dues à la superposition des signaux des autres usagers, qui sont simultanément existants dans le système, au signal de l'utilisateur d'intérêt. Dans un système CDMA, tous les usagers simultanés partagent la même bande fréquentielle et la même division temporelle. Ce principe est présenté à la figure 4.

Dans cette figure, considérons l'utilisateur k comme l'utilisateur d'intérêt. À la réception, dans une période temporelle d'observation T (période d'un symbole), qui correspond normalement à une période des codes signatures, il y a non seulement le signal de cet usager, mais aussi ceux des autres usagers qui se superposent au signal de celui-ci. Ces signaux – de l'utilisateur d'intérêt et des utilisateurs d'interférences – peuvent être synchronisés ou non synchronisés dans le temps. Dans le cas synchrone (cdma2000 par exemple), une référence temporelle comme dans les systèmes GPS doit être utilisée. Dans le cas asynchrone (WCDMA par exemple), aucune référence temporelle pour la synchronisation des signaux n'est requise. Pour la généralité, on considère un

système asynchrone dans lequel chaque usager k est caractérisé par un délai relatif τ_k . Les valeurs de ces délais varient selon les conditions d'évanouissement des canaux de communication, notamment la dispersion temporelle.

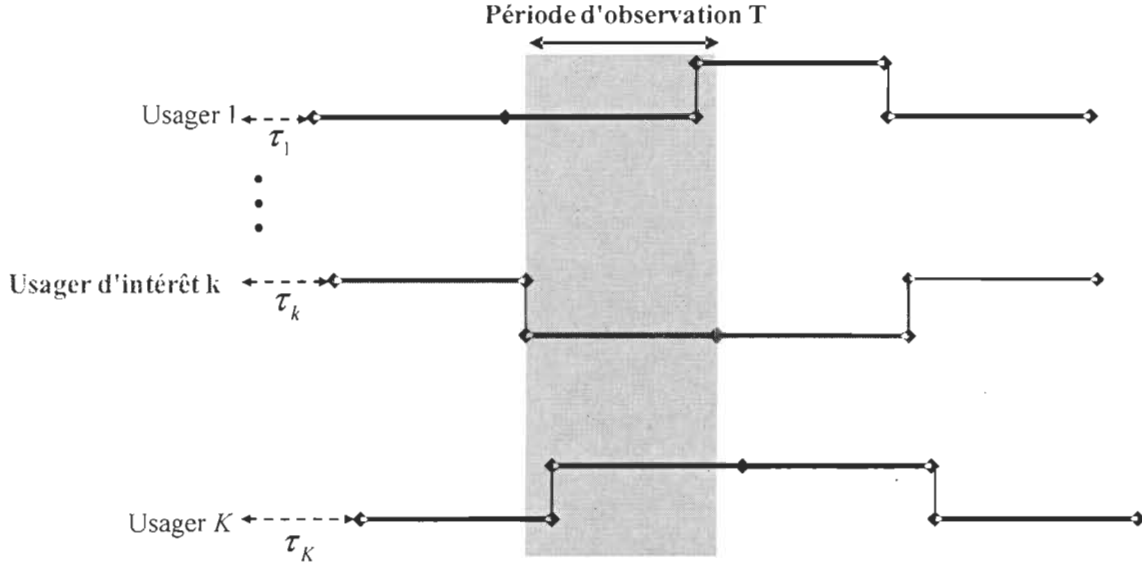


Figure 4 : Représentation des interférences à accès multiples.

Ainsi, selon les délais relatifs des usagers, le signal de l'utilisateur d'intérêt est superposé par des signaux d'interférences postérieurs ou antérieurs dans le temps par rapport à la période d'observation de celui-ci. Bien que ces délais relatifs varient d'une condition d'évanouissement à l'autre, on trouve qu'en général, pour des raisons de commodité analytique, les signaux d'interférences sont étendus partiellement sur une période postérieure, une période présente et une période antérieure [90]-[91], [109]. En général, les signaux d'interférences peuvent être considérés étendus sur un multiple des périodes postérieures et antérieures [109].

Les interférences MAI perturbent l'orthogonalité des codes signatures des usagers. Par conséquent, l'application d'un récepteur simple³ comme le *Rake*, basé sur l'orthogonalité parfaite des codes signatures des usagers, s'avère inefficace en présence des MAI.

³ Dans ce contexte, le récepteur est simple quant à l'annulation des interférences.

II.1.3. Modèle de système DS-CDMA

La figure 5 présente le modèle en bande de base d'un système DS-CDMA. Dans ce modèle, considérons K usagers transmettant des symboles de l'alphabet Ξ . Ce dernier produit les symboles $\Xi = \{-1, 1\}$ pour la modulation BPSK, et $\Xi = \left\{ \frac{1+j}{\sqrt{2}}, \frac{1-j}{\sqrt{2}}, \frac{-1+j}{\sqrt{2}}, \frac{-1-j}{\sqrt{2}} \right\}$ pour la modulation QPSK. Les symboles de chaque usager sont étalés par un code pseudo-aléatoire (*PN : Pseudo noise*) de longueur N_c . La période des symboles est dénotée comme T . La période des brides (*chips*) est dénotée comme T_c où le taux d'étalement $N_c = T/T_c$ est un entier.

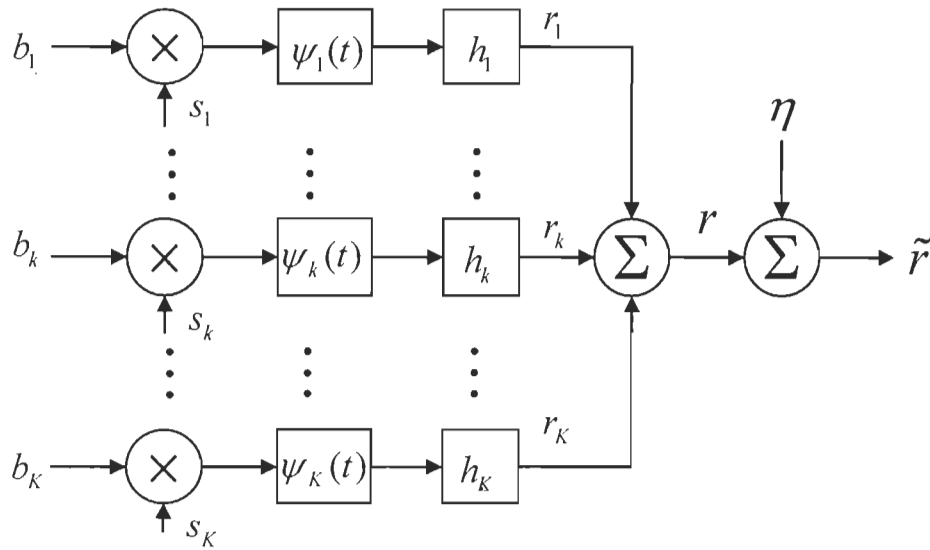


Figure 5 : Modèle en bande de base d'un système DS-CDMA. Pour simplifier la synchronisation à la réception, on considère le même filtre impulsionnel ($\psi_1(t) = \psi_k(t) = \dots = \psi_K(t)$).

Sauf indication contraire, la notation qui suit s'applique dans le reste de ce manuscrit : l'indice de temps discret pour l'espace de Baud (T) – période des symboles – et pour l'espace des brides (T_c) – période des brides – est respectivement représenté par n et m ; le symbole n de l'utilisateur k est représenté par $b_k^{(n)}$; un caractère gras dénote une quantité vectorielle ou matricielle; les paramètres estimés sont dénotés avec chapeau ($\hat{\cdot}$); $(\bullet)^H$ dénote la matrice Hermitienne; et $(\bullet)^T$ représente l'opération transposée.

La forme d'onde en temps continue de la séquence d'étalement de l'utilisateur k s'écrit comme suit :

$$s_k^{(n)}(t) = \sum_{m=1}^{N_c} s_{k,m}^{(n)} \psi(t - mT_c) \quad (2)$$

où $s_{k,m}^{(n)}$ dénote la séquence d'étalement correspondant au symbole n et la bride m de l'utilisateur k .

Dans les systèmes 3G, le code pseudo aléatoire PN se compose du code canal et du code de brouillage [2], [6]. Tous les usagers utilisent le même filtre impulsionnel $\psi(t)$ qui est limité dans l'intervalle temporel $[1, T_c)$. Les canaux de transmission peuvent être considérés comme les filtres à réponse impulsionnelle (*FIR : Finite impulse response*). Ainsi, le canal $h_k^{(n)}(t)$ de l'utilisateur k s'exprime comme suit :

$$h_k^{(n)}(t) = \sum_{l=1}^L h_{k,l}^{(n)} \delta(t - \tau_{k,l}) \quad (3)$$

où L dénote le nombre de trajets du canal de transmission.

Chaque trajet l de chaque usager k est caractérisé par un profil puissance-délai : gain $h_{k,l}^{(n)}$ et délai $\tau_{k,l}$. On considère le même nombre de trajets de propagation L pour tous les usagers. Le délai maximum est dénoté par T_m . La fonction impulsionnelle de Dirac est dénotée comme $\delta(t)$.

Au récepteur, le signal reçu consiste en une sommation de tous les signaux de tous les usagers. Ainsi, en supposant le même gain de puissance pour tous les trajets d'un même usager, le signal reçu en bande de base s'exprime comme suit :

$$\tilde{r}(t) = \sum_{n=1}^{N_b} \sum_{k=1}^K A_k b_k^{(n)} \sum_{l=1}^L h_{k,l}^{(n)} s_k^{(n)}(t - nT - \tau_{k,l}) + \eta(t) \quad (4)$$

où N_b représente le nombre de symboles à considérer ;

$\tau_{k,l}$ représente le délai relatif du trajet l de l'utilisateur k ;

A_k représente l'amplitude de l'utilisateur k ;

$\eta(t)$ représente le bruit blanc additionnel Gaussien (*AWGN – Additive white gaussian noise*) de variance σ_η^2 .

En forme matricielle, le signal reçu peut s'exprimer comme suit [30] :

$$\tilde{\mathbf{r}} = \mathbf{S}\mathbf{H}\mathbf{A}\mathbf{b} + \boldsymbol{\eta} \quad (5)$$

Considérant N_b symboles reçus avec le facteur d'étalement N_c , les matrices dans l'équation (5) sont les suivantes :

$\tilde{\mathbf{r}}$ est le vecteur des symboles bruités reçus et $\tilde{\mathbf{r}} \in \mathbb{C}^{N_b N_c \times 1}$;

\mathbf{S} est la matrice des codes PN et $\mathbf{S} \in \mathbb{C}^{N_b N_c \times KL}$;

\mathbf{H} est matrice des canaux de transmission des usagers et $\mathbf{H} \in \mathbb{C}^{KL \times KN_b}$;

\mathbf{A} est la matrice diagonale des amplitudes des signaux transmis des usagers et $\mathbf{A} \in \mathbb{R}^{KN_b \times KN_b}$;

\mathbf{b} est le vecteur des symboles transmis des usagers et $\mathbf{b} \in \mathbb{R}^{KN_b \times 1}$;

$\boldsymbol{\eta}$ est le vecteur des bruits blancs intervenant dans chaque bride du signal reçu et $\boldsymbol{\eta} \in \mathbb{R}^{N_b N_c \times 1}$.

Ces matrices sont définies comme ci-dessous.

$\tilde{\mathbf{r}}$ s'exprime comme suit :

$$\tilde{\mathbf{r}} = [\tilde{\mathbf{r}}^{T(1)}, \tilde{\mathbf{r}}^{T(2)}, \dots, \tilde{\mathbf{r}}^{T(N_b)}]^T \in \mathbb{C}^{N_b \times 1} \quad (6)$$

où $\tilde{\mathbf{r}}^{T(n)} = [\tilde{r}(nN_c + 1)T_c, \dots, \tilde{r}(n+1)N_c T_c]^T \in \mathbb{C}^{N_c \times 1}$

\mathbf{S} s'exprime comme suit :

$$\mathbf{S} = [\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \dots, \mathbf{S}^{(N_b)}]^T \in \mathbb{C}^{N_b \times 1} \quad (7)$$

où $\mathbf{S}^{(n)} = [\mathbf{s}_{1,1}^{(n)}, \dots, \mathbf{s}_{1,L}^{(n)}, \dots, \mathbf{s}_{k,1}^{(n)}, \dots, \mathbf{s}_{1,L}^{(n)}, \dots, \mathbf{s}_{K,1}^{(n)}, \dots, \mathbf{s}_{K,L}^{(n)}] \in \mathbb{C}^{L \times KL}$

$$\text{et } \mathbf{s}_{k,l}^{(n)} = \begin{cases} \mathbf{0}_{(N_b N_c - N_c) \times 1}^T, & \begin{cases} n = 1 \\ \tau_{k,l} = 0 \end{cases} \\ \left[[s_k(T_c N_c - \tau_{k,l} + 1), \dots, s_k(T_c N_c)]^T \mathbf{0}_{(N_b N_c - \tau_{k,l}) \times 1}^T \right], & \begin{cases} n = 1 \\ \tau_{k,l} > 0 \end{cases} \\ \left[\mathbf{0}_{(N_c n + \tau_{k,l}) \times 1}^T \mathbf{s}_k^T \mathbf{0}_{(N_c n - \tau_{k,l}) \times 1}^T \right], & \begin{cases} 1 < n < N_b \\ \tau_{k,l} > 0 \end{cases} \\ \left[\mathbf{0}_{(N_b N_c + \tau_{k,l}) \times 1}^T [s_k(T_c N_c - \tau_{k,l} + 1), \dots, s_k(T_c N_c)]^T \right], & \begin{cases} n = N_b \\ \tau_{k,l} > 0 \end{cases} \end{cases}, \text{ avec } \mathbf{s}_{k,l}^{(n)} \in \mathbb{C}^{N_c \times 1}$$

avec $\mathbf{s}_k = [s_k(T_c + 1), \dots, s_k(N_c + 1)T_c]^T \in \mathbb{C}^{N_c \times 1}$

\mathbf{H} s'exprime comme suit :

$$\mathbf{H} = [\mathbf{H}^{(1)}, \mathbf{H}^{(2)}, \dots, \mathbf{H}^{(N_b)}] \in \mathbb{C}^{L \times N_b} \quad (8)$$

où $\mathbf{H}^{(n)} = \text{diag}[\mathbf{h}_1^{(n)}, \mathbf{h}_2^{(n)}, \dots, \mathbf{h}_K^{(n)}] \in \mathbb{C}^{K \times K}$

et $\mathbf{h}_k^{(n)} = [h_{k,1}^{(n)}, h_{k,2}^{(n)}, \dots, h_{k,L}^{(n)}]^T \in \mathbb{C}^{L \times 1}$

\mathbf{A} s'exprime comme suit :

$$\mathbf{A} = \text{diag}[\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N_b)}] \in \mathbb{R}^{N_b \times N_b} \quad (9)$$

où $\mathbf{A}^{(n)} = \text{diag}[A_1^{(n)}, A_2^{(n)}, \dots, A_K^{(n)}] \in \mathbb{R}^{K \times K}$

\mathbf{b} s'exprime comme suit :

$$\mathbf{b} = [\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \dots, \mathbf{b}^{(N_b)}]^T \in \mathbb{R}^{N_b \times 1} \quad (10)$$

où $\mathbf{b}^{(n)} = [b_1^{(n)}, b_2^{(n)}, \dots, b_{N_c}^{(n)}]^T \in \mathbb{R}^{N_c \times 1}$

Pour l'expression mathématique qui suit ultérieurement dans ce manuscrit, on définit ici la matrice d'inter-corrélation \mathbf{R} des codes signatures comme suit :

$$\mathbf{R} = \mathbf{S}^H \mathbf{S} = \begin{bmatrix} \mathbf{R}^{(1,1)} & \dots & \mathbf{R}^{(1,D)} & \mathbf{0}_{KL} & \mathbf{0}_{KL} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{R}^{(D,1)} & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0}_{KL} & \ddots & \ddots & \ddots & \mathbf{R}^{(D-1,N_b)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0}_{KL} & \dots & \mathbf{0}_{KL} & \mathbf{0}_{KL} & \mathbf{R}^{(D,N_b)} \end{bmatrix}, \mathbf{R} \in \mathbb{C}^{KL \times KL} \quad (11)$$

Aussi, la décision sur les bits détectés se réalise par l'opération $\text{sign}(x)$ comme suit :

$$\text{sign}(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases} \quad (12)$$

II.2. Détection à usagers multiples

II.2.1. Description générale de la MUD

Il existe certaines façons de classifier les différents types de récepteurs à usagers multiples. En fait, une classification, qui se base sur la connaissance préalable des paramètres des canaux, distingue deux types d'algorithmes de MUD : méthodes indirectes et méthodes directes. Les méthodes indirectes reposent sur les paramètres déjà connus et estimés auparavant par une méthode d'identification du canal. Les méthodes directes estiment directement ces paramètres

sur lesquels elles détectent les données transmises sans besoin d'identification du canal en se basant sur des données transmises connues du récepteur.

Par une autre classification, les MUD peuvent être considérés comme centralisés ou décentralisés. Tandis que les algorithmes centralisés détectent conjointement tous les usagers en même temps – détection jointe – les algorithmes décentralisés détectent un seul usager à la fois.

La figure 6 présente une autre classification des MUD (liste non-exhaustive) selon la façon qu'ils traitent des interférences MAI, les performances, la complexité arithmétique, et la linéarité.

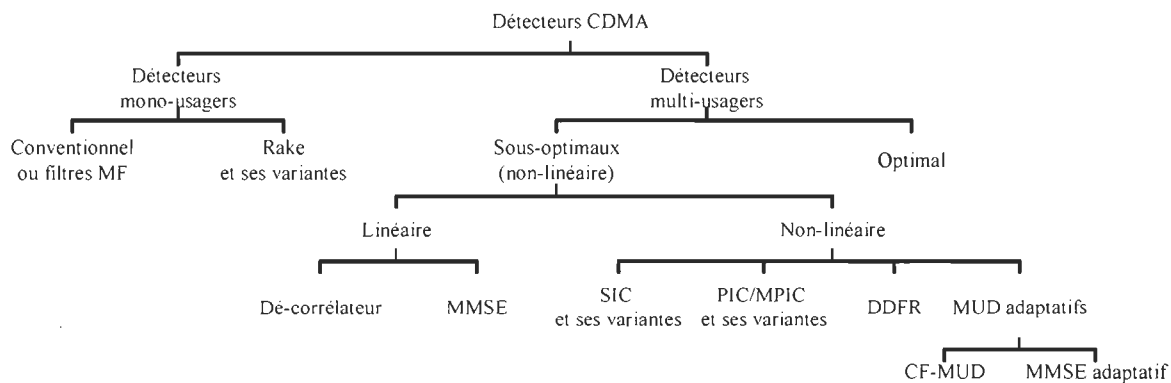


Figure 6 : Classification des récepteurs à usagers multiples (liste non-exhaustive).

Ainsi, les récepteurs mono-usagers détectent les usagers sans tenir compte de la présence des MAI, contrairement aux récepteurs à usagers multiples qui détectent les usagers tout en tenant compte de ces dernières. Par cette classification, on considère les filtres appariés et les récepteurs *Rake* comme récepteurs mono-usager. La suite de ce sous-chapitre décrit les détecteurs standards.

II.2.2. Filtre apparié

Les filtres appariés MF sont aussi connus sous le nom de récepteurs conventionnels. Il s'agit du premier récepteur utilisé dans les premiers systèmes CDMA [16], [24], [109], [110].

Le principe des récepteurs conventionnels est présenté à la figure 7. Chaque filtre MF est apparié au code d'étalement d'un usager k pour détecter ses bits tout en ignorant les interférences MAI dues aux autres usagers. Les sorties des filtres MF sont ensuite échantillonnées à la fréquence des symboles ($1/T$), alors que les opérations s'effectuent à la fréquence des brides ($1/T_c$).

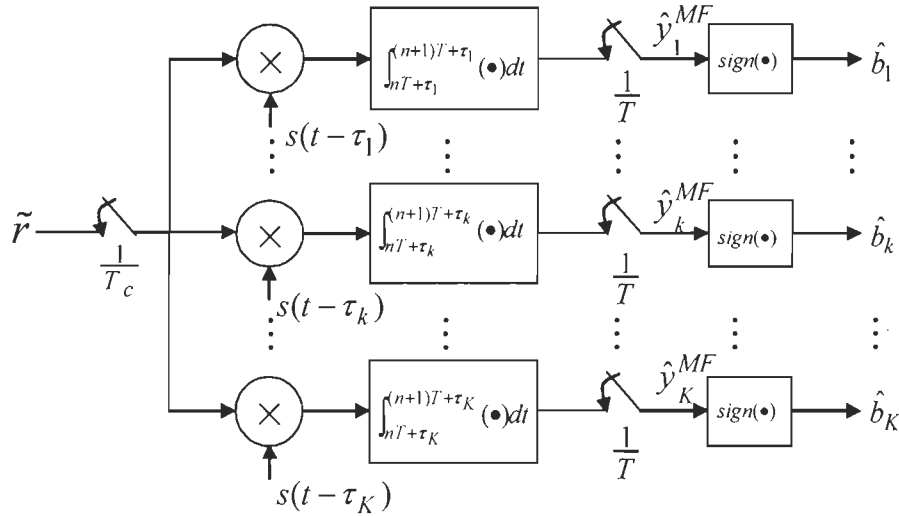


Figure 7 : Principe du filtre apparié (MF).

Le filtre MF peut être mis en équation comme suit :

$$\hat{y}_k^{MF} = \frac{1}{T} \int_{-T+\tau_k}^{T+\tau_k} r(t) s(t - \tau_k) dt \quad (13)$$

$$y_k^{MF} = A_k b_k + \sum_{i \neq k}^K \rho_{i,k} A_i b_i + \frac{1}{T} \int_{-T+\tau_k}^{T+\tau_k} \eta(t) s(t - \tau_k) dt \quad (14)$$

$$y_k^{MF} = A_k b_k + MAI + \eta_k \quad (15)$$

$$\hat{b}_k = \text{sign}(\hat{y}_k^{MF}) \quad (16)$$

où \hat{y}_k^{MF} et \hat{b}_k dénotent respectivement la sortie douce et dure du récepteur; $\rho_{i,k}$ dénote l'inter-corrélation entre le code signature de l'utilisateur i et celui de l'utilisateur k .

Le récepteur conventionnel est de faible complexité. Toutefois, les MF perturbent davantage les performances en présence des trajets multiples. En plus, ils ignorent les bruits tels que les

$$MAI = \sum_{i \neq k}^K \rho_{i,k} A_i b_i \quad \text{et le bruit de fond } \eta_r = \frac{1}{T} \int_{\tau_k}^{\tau_k+T} \eta(t) s(t - \tau_k) dt.$$

II.2.3. Récepteur *Rake* et ses variantes

Le récepteur *Rake* a été proposé afin de renforcer les performances du MF en présence d'interférences ISI qui sont dues aux trajets multiples. En effet, il se compose de plusieurs filtres MF pour exploiter la diversité spéciale de ces derniers. Chaque filtre MF estime un trajet d'un usager d'intérêt. Le principe du récepteur est présenté à la figure 8.

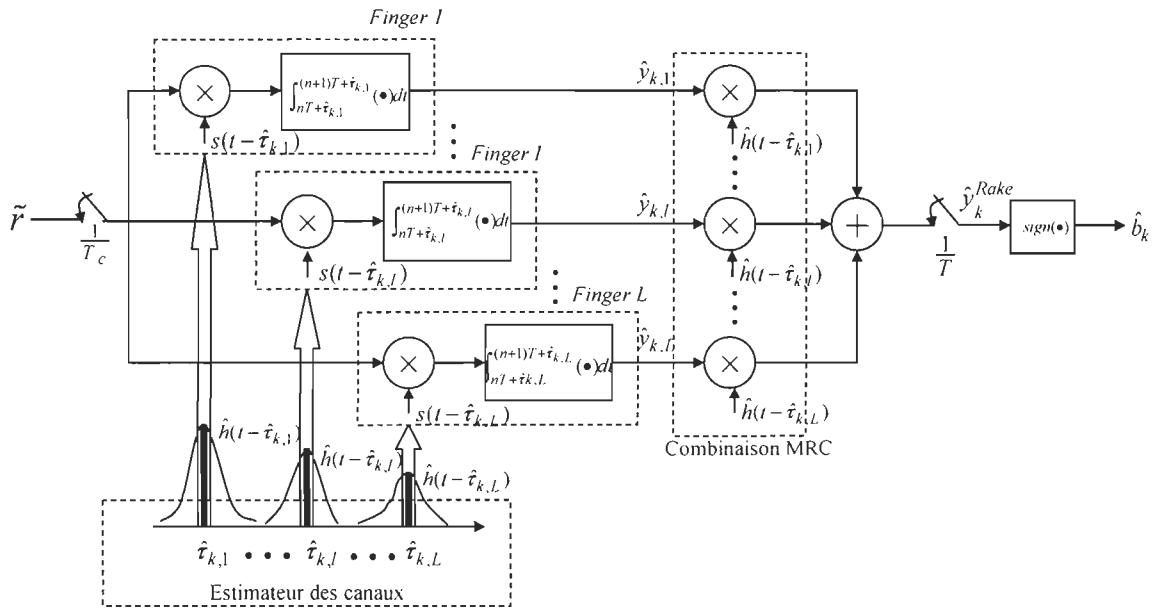


Figure 8 : Principe du récepteur *Rake* combinant L trajets de l'utilisateur k .

Le récepteur *Rake* se compose d'un estimateur des canaux et d'un nombre de filtres MF pour chaque usager. Chaque filtre MF est connu comme un doigt (*finger*) qui détecte les données d'un trajet d'un usager. Le nombre de *fingers*, ou de trajets multiples, dépend des conditions d'évanouissement du canal de communication (dispersion du canal)⁴. L'estimateur des canaux vise à estimer les paramètres des canaux. Il s'agit d'estimer les trajets dont les puissances sont les plus fortes. Ainsi, chaque trajet l de chaque usager k est caractérisé par un profil délai-puissance : le délai estimé $\hat{\tau}_{k,l}$ et le gain estimé $\hat{h}_{k,l}$.

L'estimateur des canaux a aussi pour but de trier les canaux des usagers en ordre décroissant dans le cas d'annulation successive. L'annulation successive signifie qu'un trajet d'un usager est détecté à la fois. Les sorties douces de chaque *finger* d'un usager sont ensuite combinées pour former la sortie de cet usager. Pour ce faire, la combinaison peut s'effectuer par la méthode de combinaison à ratio maximale (*MRC : Maximum ratio combining*) [19] comme présenté à la figure 8.

Le récepteur *Rake* peut être mis en équations comme suit :

$$\hat{\mathbf{y}}^{MF} = \mathbf{S}^H \mathbf{r} = \mathbf{A} \mathbf{R} \mathbf{H} \mathbf{A} \mathbf{B} + \mathbf{S}^H \boldsymbol{\eta} \quad (17)$$

$$\hat{\mathbf{y}}^{Rake} = (\mathbf{H} \mathbf{A})^H \hat{\mathbf{y}}^{MF} \quad (18)$$

où $\hat{\mathbf{y}}_k^{Rake}$ dénote la sortie douce du récepteur *Rake* et $\hat{\mathbf{y}}^{MF} \in \mathbb{R}^{KL N_b \times 1}$ et $\hat{\mathbf{y}}^{Rake} \in \mathbb{R}^{KN_b \times 1}$.

Il est à noter que ce principe représente une structure directe. Or, dans une structure indirecte, le récepteur *Rake* exclut l'estimateur des canaux. Du coup, les paramètres estimés des canaux s'effectuent par une méthode d'identification auparavant. Pour chaque usager, les sorties de ses *fingers* sont combinés avec pondération des gains des trajets respectifs.

Il existe certaines variantes du *Rake* :

⁴ Dans la pratique d'implémentation, le nombre de *fingers* considéré est déterminé par la complexité [du récepteur] qui est acceptable.

- *Rake* généralisé (*G-Rake – Generalized Rake*) [16]: Par rapport au *Rake* conventionnel, le *G-Rake* peut avoir plus ou moins de trajets et gains d'un usager à l'autre. Le *G-Rake* offre un gain de performance de 1 à 3.5 dB par rapport au *Rake* conventionnel.
- *Rake-2D* : Il renforce les performances du *Rake* conventionnel en utilisant un réseau d'antennes multiples [24]. L'utilisation d'un réseau d'antennes multiples permet de combiner les trajets multiples à la fois dans l'espace et dans le temps. Ainsi, le récepteur joue sur la diversité spatiale et temporelle.

II.2.4. Récepteur optimal

Le récepteur optimal consiste à choisir la séquence de bits transmis b de manière à maximiser la probabilité à posteriori (*MAP : Maximum a posteriori probability*) du signal bruité $r(t)$ dans l'équation (20) [109]. Autrement dit, ceci satisfait la probabilité conditionnée maximale comme suit :

$$\max_b P\{\mathbf{b} | \tilde{r}(t) : \forall t\} \quad (19)$$

Ce type de récepteur offre la performance optimale. Si toutes les séquences possibles de bits transmises possibles sont équiprobables, ce schéma de détection est le récepteur à estimation de séquence à vraisemblance maximale (*MLSE : Maximum likelihood sequence estimation*). La détection consiste à rechercher exhaustivement $2^{KL N_b}$ séquences possibles pour KL séquences (K usagers à L trajets) de N_b bits.

La complexité peut se réduire si l'on considère les sorties des filtres appariés au lieu de considérer directement le signal reçu bruité $r(t)$. Les opérations nécessaires sont ainsi d'une complexité de $O(2^{KL})$ pour cette simplification au lieu de $O(2^{KL N_b})$ [79]. Due à sa complexité énorme, le récepteur optimal est impraticable malgré ses performances. Cet inconvénient rend pertinente la recherche continue sur les récepteurs sous-optimaux (décorrélateur, MMSE, SIC, PIC, etc.) étudiés dans la suite.

II.2.5. Décorrélateur

Le décorrélateur est aussi connu comme le récepteur à forçage à zéro (ZF : *Zero forcing*). Le principe du décorrélateur est présenté à la figure 9.

Le décorrélateur se repose sur l'application de l'inversion de la matrice (\mathbf{RH}) , où \mathbf{R} et \mathbf{H} sont respectivement définis comme dans les équations (11) et (8) (cf. l'équation 20 ci-dessous). Cette dernière correspond à la convolution du canal avec les codes de chaque usager respectif. Le décorrélateur traite les interférences ISI et MAI en les forçant à zéros (donc le nom forçage à zéro) selon sa mise en équation comme suit :

$$\hat{\mathbf{y}}^{ZF} = (\mathbf{RH})^{-1} \hat{\mathbf{y}}^{Rake} \quad (20)$$

où $\hat{\mathbf{y}}^{ZF}$ dénote la sortie douce du récepteur ZF.

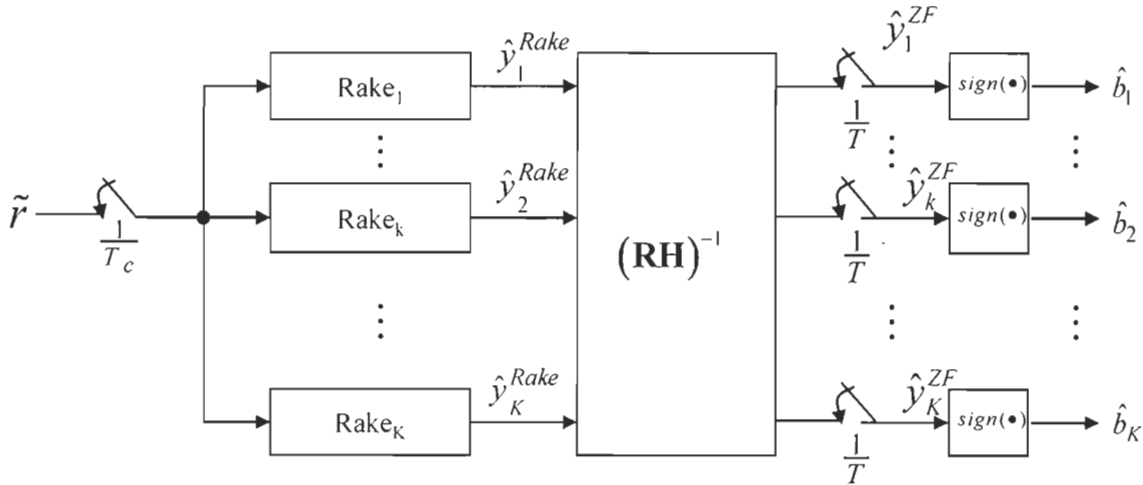


Figure 9 : Principe du décorrélateur (ZF).

II.2.6. Récepteur MMSE

Le récepteur MMSE (*Minimum mean square error*) vise à améliorer les performances du décorrélateur en tenant compte de la présence des bruits. Le principe du MMSE est présenté aux figures 10 et 11. Plus de détails mathématiques se trouvent dans [30]

Il existe deux approches pour combiner la diversité des trajets multiples, soit la combinaison des trajets multiples après (figures 10) ou avant (figures 11) le traitement du MUD.

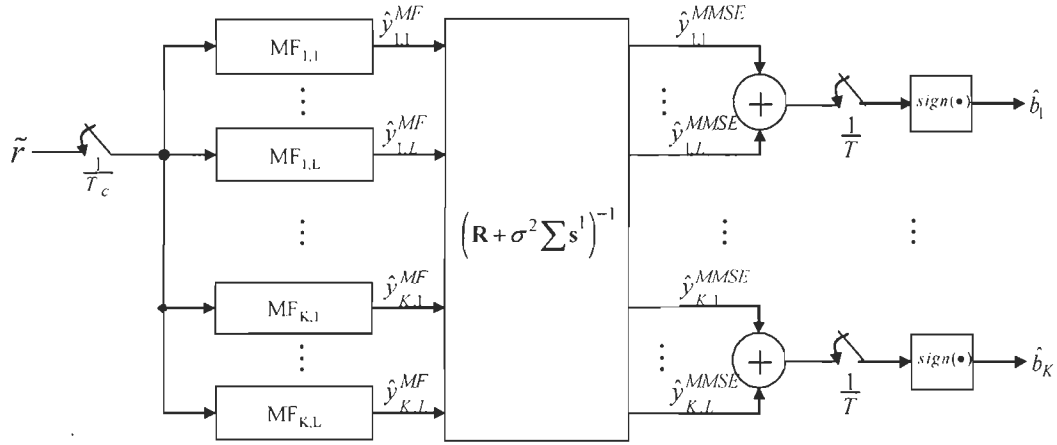


Figure 10 : Principe du récepteur MMSE avec combinaison des trajets multiples après la MUD.

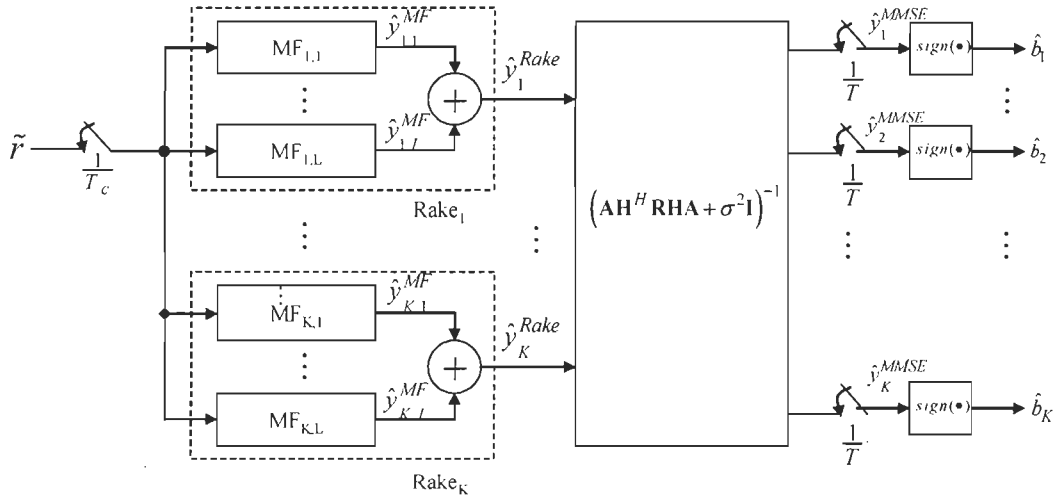


Figure 11 : Principe du récepteur MMSE avec combinaison des trajets multiples avant la MUD.

II.2.7. Récepteur SIC et ses variantes

Le récepteur SIC (*Serial interference canceller*) estime séquentiellement les données des usagers l'un avant l'autre. On peut constater qu'une fois que les données d'un usager sont détectées, on peut les utiliser pour éliminer ses interférences aux autres usagers. Au fait, les données estimées du premier usager sont utilisées, avec son amplitude et code d'étalement, pour produire (re-

moduler) un signal étalé respectif. Ce signal étalé, dérivé des données estimées du premier usager, est soustrait du signal bruité reçu pour produire le signal à détecter par le deuxième usager, et ainsi de suite jusqu'au dernier usager. Normalement, les usagers sont détectés en ordre descendant de leurs puissances respectives. Cet ordre décroissant s'explique par le fait que l'utilisateur avec la puissance plus importante introduit plus de MAI.

Ce principe du SIC se repose sur l'exactitude parfaite de l'estimation des usagers pour détecter les usagers suivant. Toutefois, les données détectées erronées peuvent éventuellement introduire une autre source de bruit [109]. Pour surmonter ce problème, le SIC se compose de plusieurs étages [7]. On a alors le SIC à étages multiples (*MSIC : Multistage SIC*). Le principe du MSIC est présenté à la figure 12.a, avec utilisation d'un *Rake* à l'entrée. La structure d'un étage est présentée à la figure 12.b. Le bloc générateur de l'utilisateur k tap l est aussi présenté à la figure 12.a.

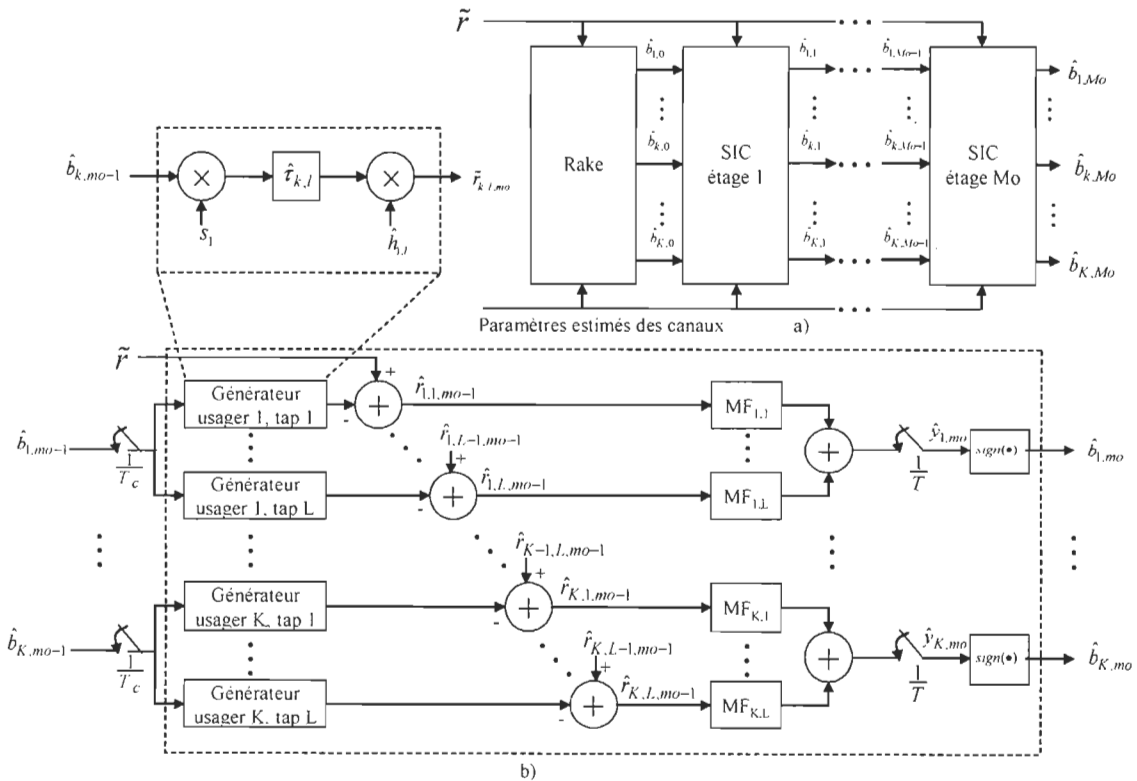


Figure 12 : Principe du récepteur SIC à Mo étage : a) Structure bloc, b) Structure d'un étage mo .

Le SIC offre certains avantages d'implémentation matérielle par sa simplification. Toutefois, la nature de la détection successive d'un usager avant les autres peut introduire une latence importante. Cette nature successive introduit une dépendance aux données [les données estimées du premier usager sont utilisées pour estimer les données du deuxième usager, puis les données de ces deux usagers sont utilisées pour estimer les données du troisième usager, ainsi de suite]. Ainsi, la dépendance aux données complique le parallélisme en implémentation matérielle. Une autre limitation [concernant la performance de la détection] du SIC se trouve dans son indépendance des paramètres estimés des canaux : délais et amplitudes [7].

II.2.8. Récepteur PIC et ses variantes

Contrairement au SIC, qui détecte séquentiellement les données des usagers, le PIC, présenté aux figures 13-14, détecte parallèlement dans le temps les données de tous les usagers. Le PIC consiste habituellement en M étage, avec entrées venant des sorties d'un *Rake*, afin de réduire les interférences résiduelles après chaque étage [107]. Par conséquent, on a le PIC à étages multiples (*MPIC : Multistage PIC*). L'implémentation du MPIC peut s'effectuer au niveau des brides (figure 13) ou au niveau des bits (figure 14).

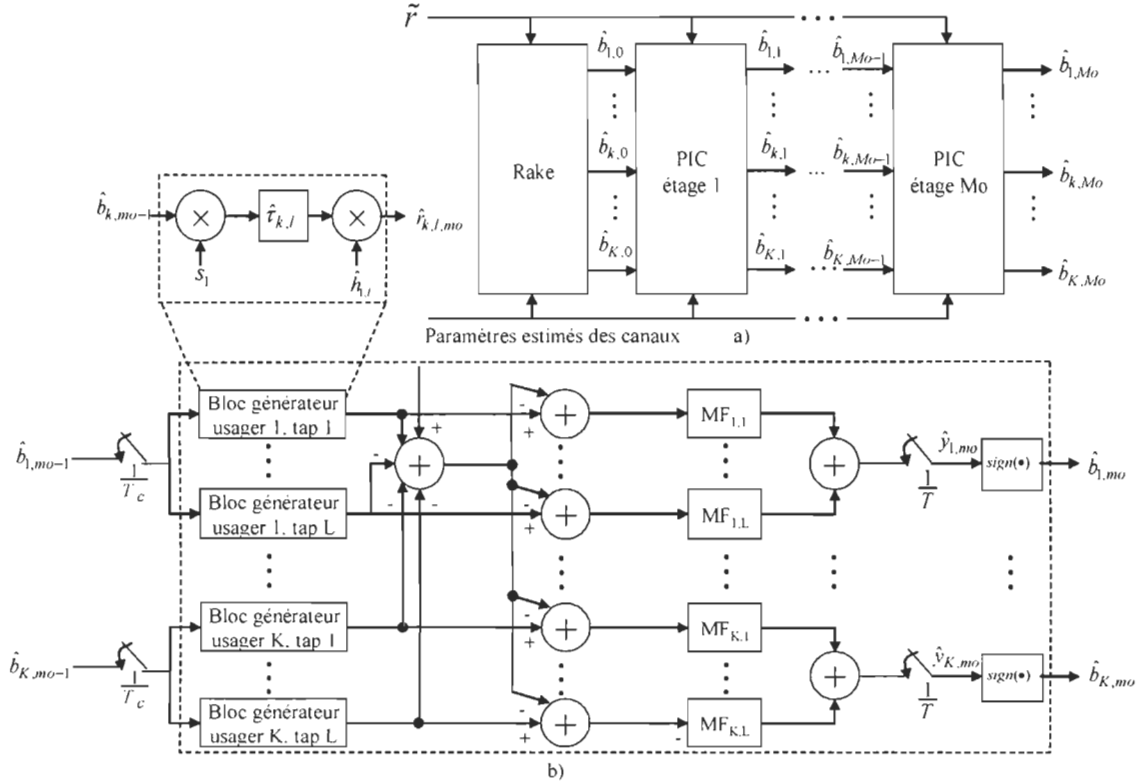


Figure 13 : Principe du récepteur MPIC à M_o étage au niveau des brides : a) Structure bloc, b) Structure d'un étage mo .

Dans ces deux figures, les sorties dures d'un bloc sont considérées. D'ailleurs, on peut considérer les sorties douces d'un étage. Du coup, les blocs de décision de signe, $sign(\bullet)$, sont exclus. Le bloc générateur de l'utilisateur k tap l et le bloc générateur des interférences sont présentés dans les deux figures.

Il existe certaines variantes du PIC :

- *Hard* MPIC : Le hard MPIC applique une fonction de décision simple – $sign(\bullet)$ – à la sortie d'un étage pour estimer un étage suivant. Il s'agit d'une structure conventionnelle du PIC comme présentée aux figures 13 et 14.

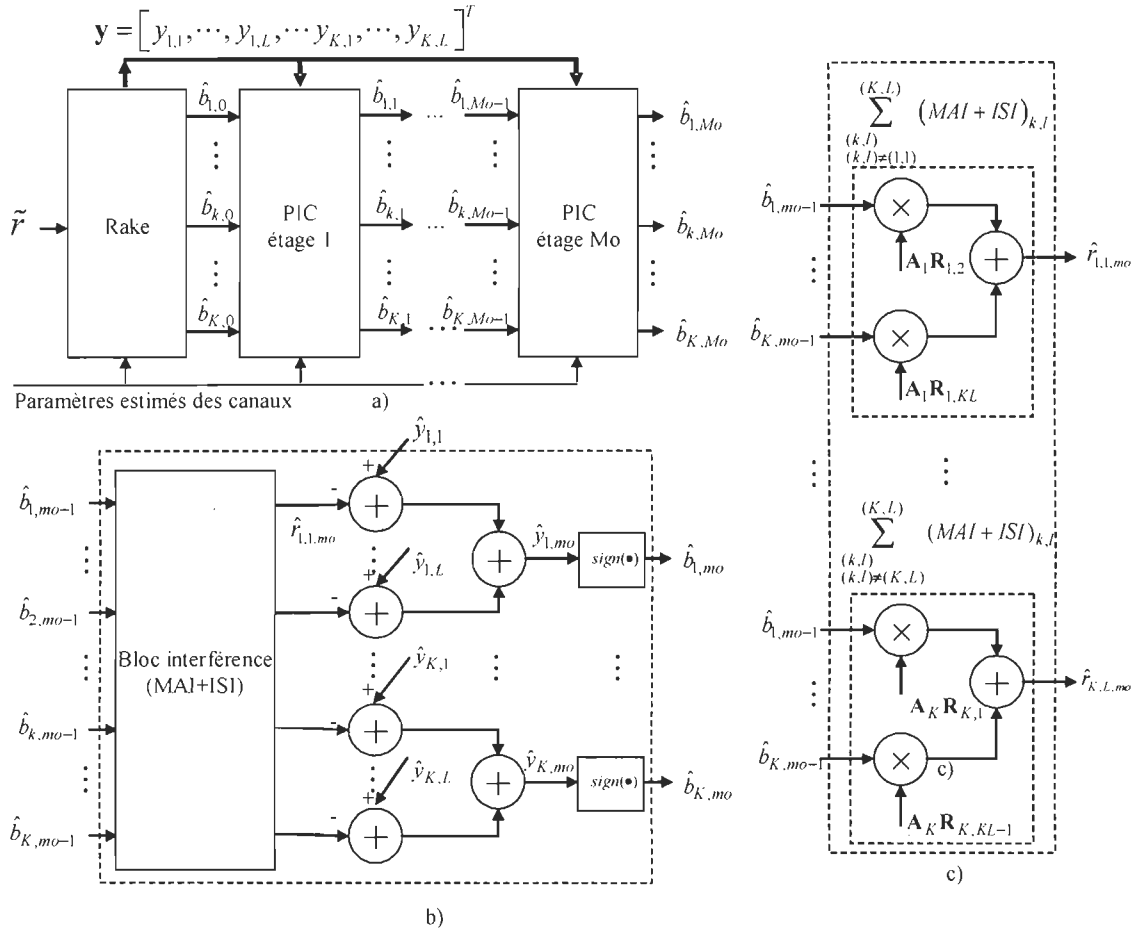


Figure 14 : Principe du récepteur MPIC à M_o étage au niveau des bits : a) Structure bloc, b) Structure d'un étage m_o .

- *Soft* MPIC : Au lieu d'appliquer la fonction $sign(\bullet)$ à la sortie de chaque trajet de chaque usager afin de produire une estimée dure (*hard output*) d'un étage pour estimer l'étage suivant, les récepteurs *soft* MPIC utilisent une fonction de décision plus complexe, par exemple la fonction de saturation, de tangente hyperbolique, etc. Une telle fonction de décision plus complexe permet de réduire une divergence éventuelle qui est produite par la fonction $sign(\bullet)$.
- *Partial* MPIC. Au lieu d'annuler directement les MAI, le MPIC partiel applique une pondération à chacun des étages par des coefficients de confiance [33].

II.2.9. Complexité des récepteurs à usagers multiples

Le tableau 2 présente la complexité des récepteurs présentés dans ce présent chapitre. Le récepteur optimal est exclu à cause de sa complexité énorme. Il s'agit d'une complexité relative en fonction des nombres d'opérations arithmétiques requises. La relativité de cette étude est due au fait que nous considérons les opérations arithmétiques en virgule flottante sur les données réelles⁵ pour tous ces récepteurs. Nous considérons, pour cette étude, un système DS-CDMA de K usagers avec un canal de L trajets multiples et des codes d'étalement de longueur N_c . Aussi, les récepteurs fonctionnent au niveau des brides.

Tableau 2 : Complexité des récepteurs à usagers multiples.

Type de MUD	Nb. d'opérations requises
Conventionnel (filtre apparié MF)	$O(KN_c)$
<i>Rake</i>	$O(LKN_c)$
Décorrélateur (ZF)	$O((KN_c)^3)$
MMSE	$O((KN_c)^3)$
SIC	$O(KN_c)$
PIC	$O(KN_c)$

Ainsi, le tableau 2 présente la complexité en ordre du nombre d'opérations nécessaires pour un symbole (de N_c brides) de données. Dans ce tableau, on considère un *Rake* comme le pré-traitement des récepteurs sous-optimaux (décorrélateur, MMSE, SIC et PIC). Aussi, on considère que la combinaison des trajets multiples s'effectue au niveau du récepteur *Rake* avant l'entrée à ces derniers récepteurs. On peut constater que le tableau 2 donne une analyse relative de la complexité sans compter du fait que le *Rake* utilise L filtres MF à l'entrée et les autres

⁵ Ce qui n'est pas toujours le cas, notamment dans les systèmes DS-CDMA de la 3G, le signal reçu est à valeurs complexes.

récepteurs (décorrélateur, MMSE, SIC, PIC) utilisent le *Rake* à l'entrée. Ainsi, le PIC et le SIC est en réalité plus complexe en tenant compte du *Rake* en soi à l'entrée. Le travail dans [20] donne une étude de complexité du décorrélateur, du PIC et SIC. Cependant, tout comme dans le tableau 2, le travail dans [20] donne également une complexité relative. En réalité, la complexité détaillée varie selon chaque implémentation qui exploite la particularité des systèmes visés pour simplifier l'arithmétique des détecteurs d'intérêt comme nous verrons dans la section II.4.

II.3. Détection à usagers multiples multi-débit

La communication sans fil multi-débit, c.-à-d. où il existe en même temps plusieurs débits dans un même système de communications, permet de transmettre à la fois plusieurs données de débits différents. Plusieurs techniques possibles permettent des systèmes de communications CDMA multi-débit. Parmi ces méthodes, deux méthodes de débits multiples sont adoptées par la norme de la 3G : méthode basée sur les multicodes (*MC : Multicode*) et méthode basée sur les facteurs d'étalement variables (*VSF : Variable spreading facteur*). Les systèmes MC reposent sur l'utilisation des codes signatures d'une seule longueur, alors que les systèmes VSF reposent sur l'utilisation des codes signatures de longueurs variables dans lesquels chaque débit s'associe à une longueur de codes. Dans les systèmes multi-débit MC, l'utilisateur de haut débit est converti en plusieurs usagers virtuels de débit faible, chacun s'associe à un code signature différent. Les données associées à ces usagers virtuels sont transmises en parallèle. Une description plus détaillée sur la méthode MC peut être consultée dans [25]. Nous considérons les systèmes multi-débit basés sur la méthode VSF pour leur simplicité vis-à-vis des systèmes MC. En plus, les travaux expérimentaux dans [76] montrent que les deux méthodes MC et VSF peuvent être utilisées pour des systèmes multi-débit avec les mêmes performances dans certaines circonstances.

Cette partie vise à synthétiser les méthodes multi-débit couramment utilisées dans la littérature. En plus, nous développons un modèle cohérent avec nos travaux dans le chapitre III.

II.3.1. Modèle de système DS-CDMA multi-débit

Considérons un système multi-débit dans lequel il existe simultanément G groupes de débits de données différents comme présenté à la figure 15. Chaque groupe de débit (ou groupe des usagers à ce débit) se caractérise par un facteur d'étalement variable orthogonal (OVSF : *Orthogonal VSF*). Plus le facteur est court, plus le débit respectif est élevé. Sans perte de généralité, le groupe de débit le plus faible et le plus élevé est respectivement dénoté par exposant 1 et G en exposant.

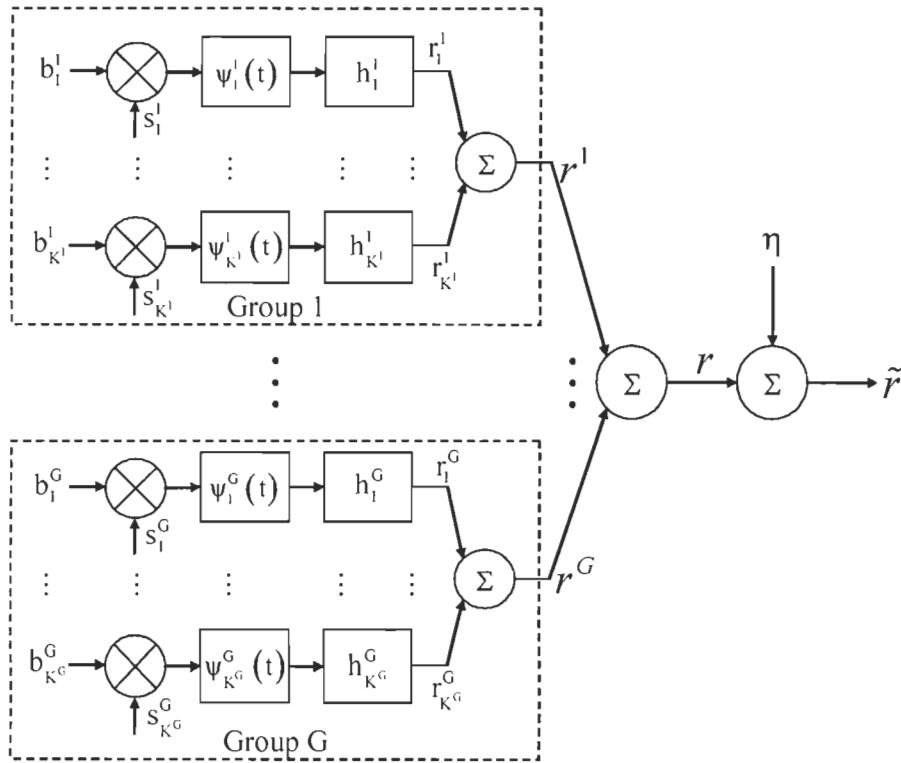


Figure 15 : Modèle en bande de base d'un système CDMA multi-débit, avec le même filtre impulsionnel $\psi_1^1(t) = \dots = \psi_{K^1}^1(t) = \dots = \psi_1^G(t) = \dots = \psi_{K^G}^G(t) = \psi(t)$.

Sauf indication contraire, la notion qui suit s'applique aux systèmes multi-débit. Le ratio M dénote le rapport entre le facteur OVSF le plus long et celui le plus court. L'exposant g dénote le groupe de débit. N_c^g et T^g dénotent respectivement la longueur et la période temporelle des codes OVSF du groupe g . L'indice k dénote le k -ième usager. L'indice i dénote la i -ième bride.

L'exposant n dénote le n -ième symbole. Le caractère s dénote le code signature. Le caractère h dénote le canal de transmission. L'indice l dénote le l -ième trajet du canal. Aussi, la notion combinaison s'utilise couramment dans les présentations mathématiques d'un système multi-débit.

En plus, par notre définition, un usager réel est un usager qui existe physiquement dans le système multi-débit considéré ; un usager virtuel est un usager qui n'existe pas physiquement dans le système multi-débit considéré mais qui est conventionnellement dérivé par une méthode multi-débit ; un usager équivalent est un usager considéré par une méthode multi-débit qui peut être réel ou virtuel. Finalement, la notion virtuelle – usager virtuel, symbole virtuel, etc. – est dénotée par une prime (').

Pour faciliter la synchronisation à la réception, on considère le même filtre impulsionnel ($\psi_1(t) = \dots = \psi_{K^1}^1(t) = \dots = \psi_1^G(t) = \dots = \psi_{K^G}^1(t) = \psi(t)$). La figure 16 présente la représentation des symboles étalés multi-débit. Dans cette figure, le débit du groupe 1 est respectivement deux fois et quatre fois plus lent que celui du groupe 2 et du groupe G . Ainsi, la largeur des codes d'étalement du groupe 1 est deux fois et quatre fois plus longue que celle du groupe 2 et du groupe G .

Dans chaque groupe g , il y a K^g usagers réels. Ainsi, le nombre d'usagers réels K dans le système est égal à la somme de tous les usagers réels de tous les groupes de débits :

$$K = \sum_{g=1}^G K^g \quad (21)$$

Le code signature, qui s'associe avec le symbole n de chaque usager k de chaque groupe g s'exprime comme suit :

$$s_k^{g(n)}(t) = \sum_{i=1}^{N_c^g} s_{k,i}^{g(n)} \psi(t - iT_c) \quad (22)$$

Supposons que les signaux des usagers se propagent à travers le même nombre de trajets (L) dans lesquels chaque trajet l de l'utilisateur k du groupe g est respectivement caractérisé par un délai relatif $\tau_{k,l}^g$ et un gain $h_{k,l}^{g(n)}$. Ainsi, le canal de chaque usager k du groupe g s'exprime comme suit :

$$h_k^{g(n)}(t) = \sum_{l=1}^L h_{k,l}^{g(n)} \delta(t - \tau_{k,l}^g) \quad (23)$$

où $h_{k,l}^{g(n)}$ dénote le gain du trajet l de l'utilisateur k du groupe g .

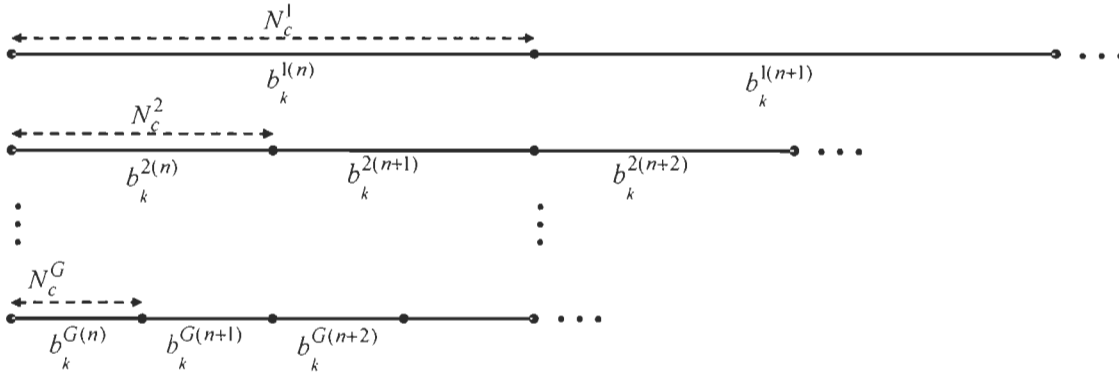


Figure 16 : Représentation des symboles étalés multi-débit, par exemple le débit du groupe 1 est deux fois et quatre fois plus lent que le groupe 2 et le groupe G .

Le signal reçu après le canal de transmission de l'utilisateur k du groupe g consiste en tous les trajets de cet usager et s'exprime comme suit :

$$r_k^g(t) = \sum_{n=1}^{N_k^g} A_k^g b_k^{g(n)} \sum_{l=1}^L h_{k,l}^{g(n)} s_k^{g(n)}(t - nT^g - \tau_{k,l}^g) \quad (24)$$

où N_k^g est le nombre de bits transmis considérés du groupe g ($b_k^{g(n)}$); A_k^g est l'amplitude de l'utilisateur k en supposant un contrôle parfait de la puissance; et T^g est la période temporelle des symboles du groupe g .

Le signal reçu du groupe de débit g consiste en une somme de tous les signaux de ce groupe comme suit :

$$r^g(t) = \sum_{k=1}^{K^g} r_k^g(t) \quad (25)$$

Le signal reçu au récepteur consiste en la somme de tous les signaux de tous les usagers de tous les groupes de débits de données. Ainsi, le modèle général du signal reçu d'un système multi-débit à la réception s'exprime comme suit :

$$\tilde{r}(t) = \sum_{g=1}^G \left\{ \sum_{n=1}^{N_k^g} \left[\sum_{k=1}^{K^g} \left(A_k^g b_k^{g(n)} \sum_{l=1}^{L_g} h_{k,l}^{g(n)} s_k^{g(n)} \left(t - nT^g - \tau_{k,l}^g \right) \right) \right] \right\} + \eta \quad (26)$$

II.3.2. Méthodes de détection à usagers multiples multi-débit

Les méthodes communes de détection multi-débit reposent sur la conversion d'un système multi-débit en un système mono-débit équivalent afin d'appliquer les méthodes de détection à usagers multiples existantes [52], [55]-[56], [77]-[78], [93]-[94], [115], [119]. En effet, on peut trouver trois méthodes multi-débit :

- La méthode basée sur un seul débit le plus faible (*LRD : Low rate detector*) (cf. II.6.2.A) [52], [55]-[56], [77]-[78], [93], [115], [119],
- La méthode basée sur un seul débit le plus élevé (*HRD : High rate detector*) (cf. II.6.2.C) [52], [55], [77], [78], et
- La méthode de détection successive (cf. II.6.2.C) [106], [111].

En plus, un changement de débit de données peut survenir au cours de la transmission. Un tel changement peut résulter en un système multi-débit. En fait, un récepteur MMSE adaptatif a été développé pour permettre un changement de taux de données [56].

II.3.2.A. Méthode LRD

Cette méthode repose sur la conversion de tous les débits du système multi-débit en un système mono-débit basé sur le débit le plus faible. En effet, un usager réel à $OVSF = N_c^g$ est converti en

N_c^1/N_c^g usagers virtuels où N_c^1 est le facteur OVSF du groupe de débit le plus faible. Dans une période observée au récepteur (T^l – période la plus longue des symboles), il y a un symbole réel de chaque usager réel du groupe de débit le plus faible et N_c^1/N_c^g symboles réels de chaque usager réel du groupe g . Donc, N_c^1/N_c^g symboles virtuels appropriés sont dérivés dans le système mono-débit équivalent. Ces symboles virtuels sont associés aux usagers virtuels correspondants.

La dérivation des usagers virtuels est présentée à la figure 17. Le débit du groupe g est quatre fois plus élevé que celui le plus faible. Dans une période des symboles à débit le plus faible T^1 , il y a un symbole réel de l'utilisateur à débit le plus faible et quatre symboles réels de l'utilisateur du groupe de débit g (figure 17.a). Ainsi, quatre usagers virtuels sont dérivés dont chacun est associé à un symbole virtuel correspondant (figure 17.b). Les zéros sont remplis pour étendre à toute la période T^1 . De cette manière, dans une fenêtre d'observation de N_b symboles,

il y a $N_{LRD} = N_b \sum_{g=1}^G N_c^1/N_c^g$ symboles équivalents pour la méthode LRD. Ainsi, on définit le

vecteur des symboles à détecter comme suit :

$$\mathbf{b} = \left[\mathbf{b}^{T^1(1)}, \mathbf{b}^{T^1(2)}, \dots, \mathbf{b}^{T^1(N_b)} \right]^T \in \mathbb{R}^{K_{LRD} N_{LRD} \times 1} \quad (27)$$

où $\mathbf{b}^{T^1(n)} \in \mathbb{R}^{K_{LRD} \times 1}$ est défini comme suit :

$$\mathbf{b}^{T^1(n)} = \left[b_1^{1(n)}, \dots, b_{K^1}^{1(n)}, \dots, b_{1,1}^{g(n)}, \dots, b_{1, \frac{N_c^1}{N_c^g}}^{g(n)}, \dots, b_{K^g, 1}^{g(n)}, \dots, b_{K^g, \frac{N_c^1}{N_c^g}}^{g(n)}, \dots, b_{K^G, 1}^{G(n)}, \dots, b_{K^G, \frac{N_c^1}{N_c^g}}^{G(n)} \right]^T \quad (28)$$

Le nombre d'utilisateurs total (K_{LRD}) comprend tous les utilisateurs équivalents – utilisateurs réels du groupe de débit le plus faible et utilisateurs virtuels dérivés des utilisateurs réels de débits élevés – du système mono-débit équivalent. Ainsi, le nombre d'utilisateurs total (K_{LRD}) s'exprime comme suit :

$$K_{LRD} = \sum_{g=1}^G \frac{N_c^1}{N_c^g} K^g \quad (29)$$

On prend, par exemple, un système à trois débits correspondant aux facteurs OVFS de 16, 8 et 4. Le nombre d'usagers réels est 8, 4 et 2 respectivement. Donc, le nombre d'usagers total par cette méthode LRD devient $K_{LRD} = \frac{16}{16}8 + \frac{16}{8}4 + \frac{16}{4}2 = 24$ incluant 8 usagers réels à OVFS=16, 8 usagers virtuels dérivés à partir des usagers réels à OVFS de 8 et 8 usagers virtuels à partir des usagers réels à OVFS de 4.

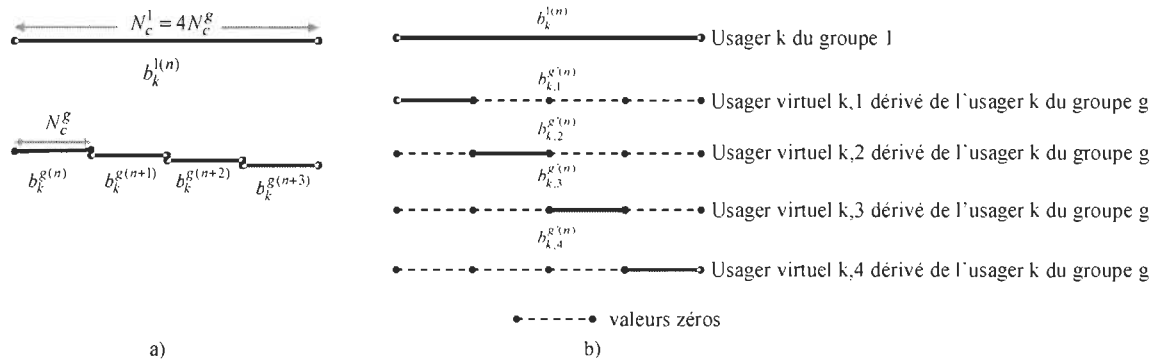


Figure 17 : Dérivation des usagers virtuels par la méthode LRD : a) symboles de l'utilisateur k du groupe g , par exemple le débit du groupe g est quatre fois plus élevé que le groupe 1, b) symboles virtuels dérivés.

Dans le système mono-débit équivalent, le signal reçu consiste en la somme de tous les usagers équivalents et s'exprime comme suit :

$$r(t) = \sum_{n=1}^{N_b} \left\{ \sum_{g=1}^G \left[\sum_{k=1}^{K^g} \sum_{v=1}^{\frac{N_c^1}{N_c^g}} r_{k,v}^{g'(n)}(t) \right] \right\} \quad (30)$$

où l'élément $r_{k,v}^{g'(n)}(t)$ est le signal transmis de l'utilisateur virtuel v qui est dérivé de l'utilisateur k du groupe g .

L'élément $r_{k,v}^{g(n)}(t)$ s'introduit car, dans cette méthode multi-débit, le n -ième symbole du k -ième usager du groupe g est converti en N_c^1/N_c^g symboles virtuels de débit le plus faible. Les symboles virtuels s'expriment comme suit :

$$b_{k,v}^{g(n)}(t) = \begin{cases} b_k^{g(n)}(t); & bT_c^g \leq t \leq (v+1)T_c^g \text{ avec } 1 \leq v \leq \frac{N_c^1}{N_c^g} \\ 0; & \text{ailleurs} \end{cases} \quad (31)$$

Où $v = 1, \dots, \frac{N_c^1}{N_c^g}$.

Avec le modèle décrit dans cette section, la MUD multi-débit s'effectuera comme pour le système mono-débit de K_{LRD} usagers à un seul débit qui est le plus faible en utilisant un algorithme mono-débit existant comme présentée à la figure 18.

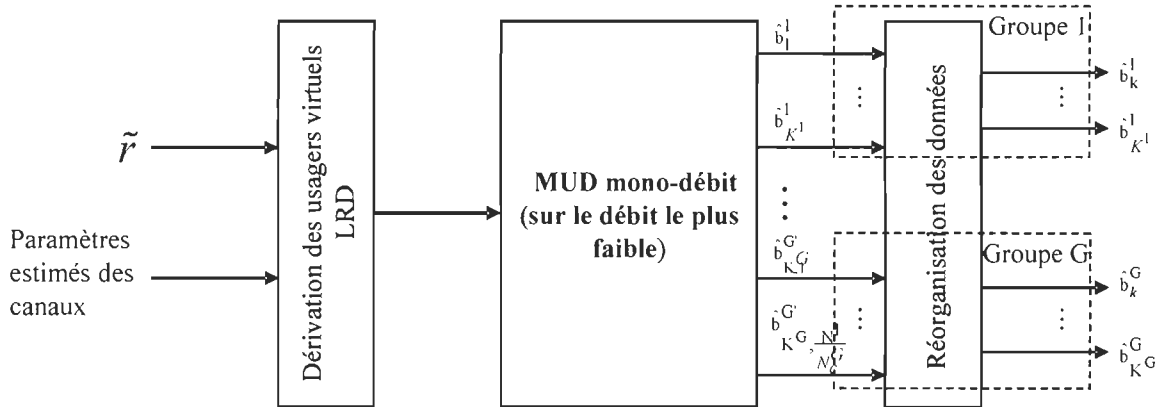


Figure 18 : Principe de la méthode LRD basée sur un MUD mono-débit.

La réorganisation des données permet d'obtenir les données détectées de chaque usager réel de chaque groupe. La réorganisation des données s'effectue sur les sorties dures des estimés. Alors que le groupe de débit le plus faible échantillonne les sorties estimées à chaque période temporelle T^1 , les autres groupes de débits plus élevés g échantillonnent les sorties estimées à chaque période temporelle respective T^g . Autrement dit, si le système est cadencé à la fréquence

correspondant à la période T^1 , il y a un symbole détecté de chaque groupe de débit le plus faible et N_c^1/N_c^g symboles détectés de chaque groupe de débit plus élevé.

II.3.2.B. Méthode HRD

Cette méthode repose sur la conversion de tous les débits du système multi-débit en un système mono-débit basé sur le débit le plus élevé. Le système équivalent est alors considéré comme celui à un seul débit le plus élevé. En fait, un usager réel à $OVSF = N_c^g$ est converti en un seul usager virtuel mais avec N_c^g/N_c^G symboles virtuels, où N_c^G est le facteur OVSF du groupe de débit le plus élevé. Dans une période observée au récepteur (T^G – période la plus courte des symboles), il y a un symbole réel de chaque usager réel du groupe de débit le plus élevé et une partie (N_c^g/N_c^G) du symbole réel de chaque usager réel du groupe g . Donc, N_c^g/N_c^G symboles virtuels appropriés sont dérivés dans le système mono-débit équivalent.

La dérivation des usagers virtuels est présentée à la figure 19. Par exemple, le débit du groupe g est quatre fois plus faible que celui le plus élevé. Dans une période des symboles à débit le plus élevé T^G , il y a un symbole réel de l'utilisateur réel à débit le plus élevé et un quart des symboles réels de l'utilisateur du groupe de débit g (figure 19.a). Donc, dans le système équivalent, un symbole virtuel dérivé de l'utilisateur k du groupe g , dont la valeur appartient à une partie d'un symbole original dans une période T^G , est formé (figure 19.b). De cette manière, dans une fenêtre d'observation de N_h symboles, il y a $N_{HRD} = N_h$ symboles équivalents pour la méthode HRD. Ainsi, on définit le vecteur des symboles à détecter comme suit :

$$\mathbf{b} = [\mathbf{b}^{T^G(1)}, \mathbf{b}^{T^G(2)}, \dots, \mathbf{b}^{T^G(N_h)}]^T \in \mathbb{R}^{K_{HRD} N_{HRD} \times 1} \quad (32)$$

où $\mathbf{b}^{T^G(n)} \in \mathbb{R}^{K_{HRD} \times 1}$ est défini comme suit :

$$\mathbf{b}^{T^G(n)} = [b_1^{1(n)}, \dots, b_{K^1}^{1(n)}, \dots, b_1^{g(n)}, \dots, b_{K^g}^{g(n)}, \dots, b_1^{G(n)}, \dots, b_{K^G}^{G(n)}]^T \in \mathbb{R}^{K_{HRD} \times 1} \quad (33)$$

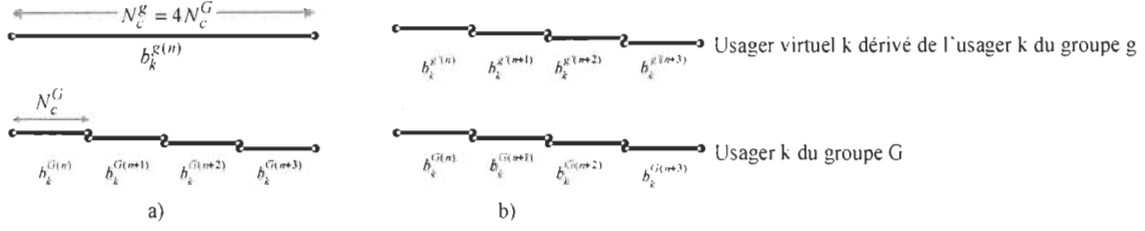


Figure 19 : Dérivation des usagers virtuels par la méthode HRD : a) Symboles de l'utilisateur k du groupe g , par exemple le débit du groupe g est quatre fois plus faible que le groupe G , b) Symboles virtuels dérivés.

Contrairement à la méthode LRD, la méthode HRD n'introduit pas d'utilisateurs supplémentaires dans le système mono-débit équivalent. Ainsi, le nombre des utilisateurs total K_{HRD} consiste en tous les utilisateurs présents dans une période T^G , et s'exprime comme suit :

$$K_{HRD} = \sum_{g=1}^G K^g \quad (34)$$

Le signal reçu au récepteur est la somme de tous les utilisateurs et s'exprime comme suit :

$$r(t) = \sum_{m+v=1}^{N_h} \left\{ \sum_{g=1}^G \left[\sum_{k=1}^{K^g} \left(r_k^{g'(m+v)}(t) \right) \right] \right\} \quad (35)$$

avec $\begin{cases} m+v=n; g=G \\ m = \text{mod}(n, \frac{T_c^g}{T^G}), g \neq G \end{cases}$, et $1 \leq v < \frac{T_c^g}{T^G}$

où l'élément $r_k^{g'(n+v)}(t)$ est le signal transmis, qui est dérivé de l'utilisateur k du groupe g , associé au symbole virtuel $b_k^{g'}(n+v)$ correspondant.

L'élément $r_k^{g'(n+v)}(t)$ s'introduit car, dans cette méthode HRD, les symboles virtuels sont dérivés du symbole n de l'utilisateur k du groupe de débit g dans une période temporelle T^G . Les symboles virtuels s'expriment comme suit :

$$b_k^{g'(m+v)}(t) = b_k^{g(m)}(t), mT^G \leq t \leq (m+v)T^G \text{ avec } 1 \leq v < \frac{T_c^g}{T^G} \quad (36)$$

Avec le modèle décrit dans cette section, la MUD multi-débit s'effectuera comme pour le système mono-débit de K_{HRD} usagers à un seul débit le plus élevé en utilisant un algorithme mono-débit comme présente à la figure 20.

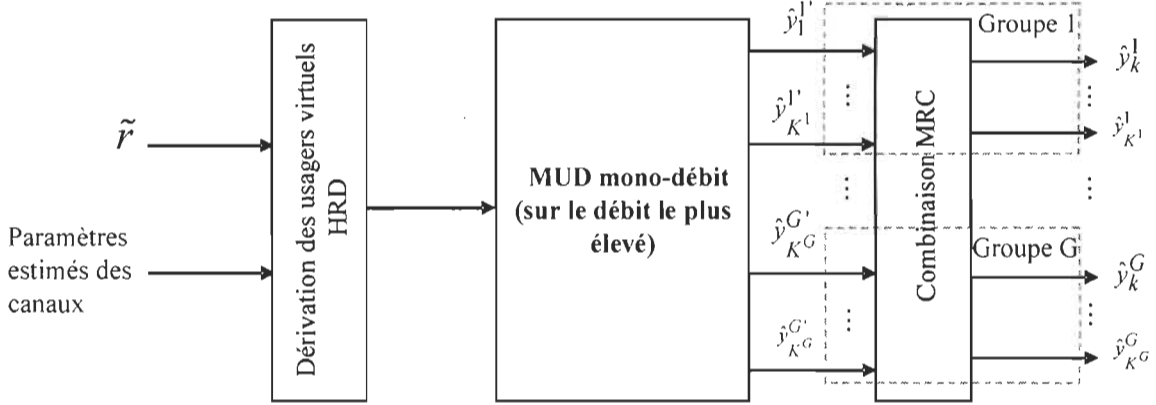


Figure 20 : Principe de la méthode HRD basée sur un MUD mono-débit.

Pour les usagers du groupe de débit le plus élevé, on échantillonne directement les sorties des données correspondantes. Toutefois, pour les usagers de débits faibles, étant donné qu'à chaque période temporelle des symboles de débit le plus élevé T^G , une seule partie des symboles est détectée, il faut alors combiner les sorties des données détectées. La méthode MRC [17] peut s'appliquer pour effectuer la combinaison des sorties douces de ces données estimées, et s'exprime comme suit :

$$\hat{y}_k^g(m) = \sum_{v=1}^{\frac{N_c^g}{N^G}} \hat{y}_k^{g'}(m+v) \quad (37)$$

où $\hat{y}_k^g(m)$ est le symbole estimé m de l'utilisateur réel k du groupe g , $\hat{y}_k^{g'}(m+v)$ est le symbole virtuel estimé, qui est dérivé du symbole $b_k^g(m)$.

II.3.2.C. Méthode successive

Le principe de la méthode multi-débit successive est présenté à la figure 21.

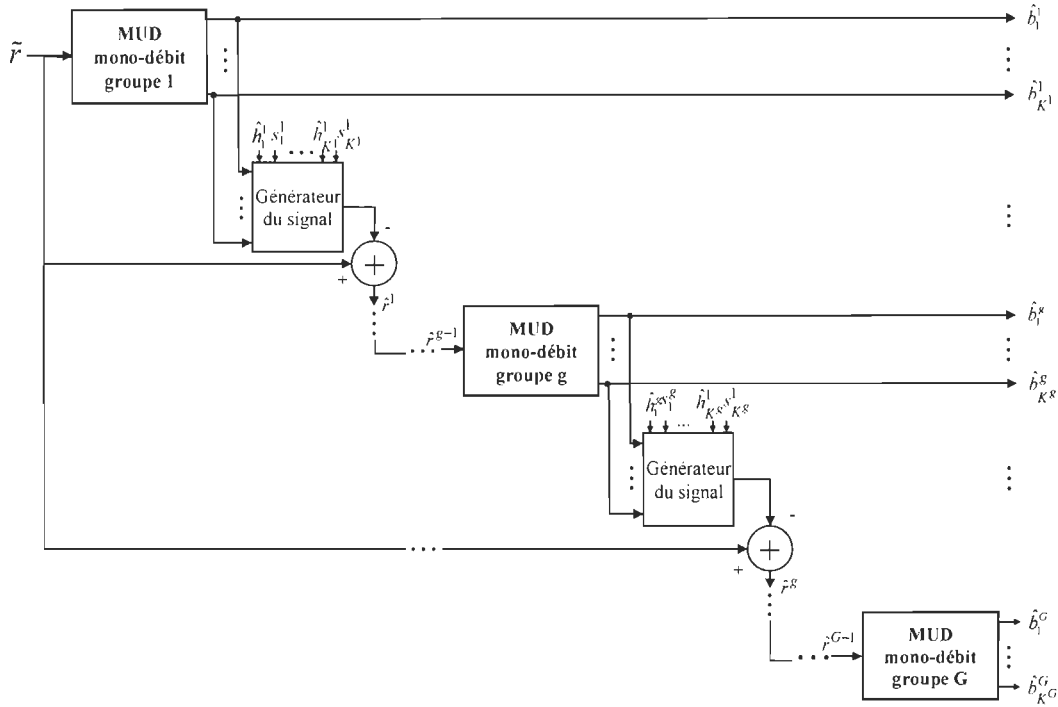


Figure 21 : Principe de la méthode multi-débit successive basée sur un MUD mono-débit.

Les travaux dans [106], [111] développent les récepteurs multi-débit qui annulent successivement à la fois les interférences pour chaque groupe de débit. Ce principe ressemble à celui utilisé dans les récepteurs mono-débit SIC qui annulent pour chaque usager d'intérêt à la fois les interférences dues aux autres usagers simultanés (cf. la section II.2.7). Les données des autres groupes de débits sont considérées comme bruit blanc au groupe de débit d'intérêt.

II.4. Travaux d'implémentation des récepteurs à usagers multiples

Le progrès de l'industrie du semi-conducteur rend possible ce qu'on connaît aujourd'hui comme étant la technologie radio logicielle (*software-defined radio*). En effet, aujourd'hui les récepteurs de communication les plus sophistiqués sont implantables par la technique dite de radio logicielle, en utilisant un DSP ou un FPGA, ou un hybride FPGA et DSP. D'un point de vue technique, la radio logicielle connaît un grand avantage principalement concernant la reconfiguration et le temps de développement vis-à-vis d'une implémentation de type

d'application spécialisée (ASIC, ASSP, etc.). En plus, la technologie de logique programmable, i.e. FPGA, nous permet la possibilité d'implémenter des systèmes-sur-puce programmable (*SOPC : System on programmable chip*) complexes. Toutefois, l'inconvénient de la technologie radio logicielle se trouve dans la consommation d'énergie, la quantité de ressources et la vitesse d'exécution par rapport à une implémentation de type d'application spécifique.

Nous allons donc citer les travaux d'implémentation les plus pertinents aussi bien sur les technologies de radio logicielle qu'à applications spécifiques.

Tenant compte de sa simplicité d'implémentation et son application dans les systèmes 2G actuels, plusieurs travaux d'implémentation du récepteur *Rake* ont été effectués [36], [54], [63], [72]. Le *Rake* est assez simple à implémenter dans un DSP [54], [72]. En plus, les DSP sont beaucoup utilisés pour implémenter les fonctions de communication dans les combinés (*handsets*) de la 2G. Le *Rake* a été implémenté sur les bancs d'essai (*testbed*) avec un composant FPGA de faible complexité et à faible fréquence d'horloge (XC4028 à 10 MHz) [63], ainsi que sur un ASIC de faible consommation [36].

Cependant, le *Rake* s'avère inefficace pour annuler des interférences MAI dans les systèmes DS-CDMA de la 3G. Comme mentionné à la section II.2.4, Verdú a proposé le MUD optimal [109], qui offre des performances optimales pour l'annulation des MAI. Toutefois, ce type de MUD n'est pas convenable pour l'implémentation en temps réel à cause de sa complexité arithmétique énorme. Ainsi, les MUD sous-optimaux, dont le but est de réduire la complexité arithmétique, ont été proposés [cf. II.2]. De plus, la plupart des algorithmes complexes tels que MMSE, dé-corrélateurs, se basent sur l'inversion des matrices. Or, l'inversion des matrices représente un surplus en complexité matérielle : complexité arithmétique en nombre de multiplications et d'additions, et nombre de bits pour assurer la précision arithmétique. Nous pouvons noter que tous les algorithmes de MUD sont basés sur l'opération sur des vecteurs et/ou des matrices. Bien que les DSP les plus récents à structure de VLIW (*Very long instruction word*) puissent être exploités, ils s'avèrent insuffisants pour l'exécution en temps réel par un seul DSP. En plus, l'implémentation efficace des MUD complexes se compose de plusieurs DSP ou d'un hybride de plusieurs DSP et FPGA [89]. En effet, les implémentations

radio logicielle utilisant un DSP ont été effectuées pour les récepteurs plus complexes, par exemple un SIC sur DSP [21], un PIC sur DSP [28].

Les implémentations en ASIC ont été proposées pour les MUD complexes tels que PIC [90], [114] et SIC [85]. Évidemment, ces implémentations ASIC offrent les avantages en vitesse d'exécution et en faible consommation par rapport à celles basées sur les DSP. Or, les implémentations ASIC perdent en flexibilité d'implémentation. D'ailleurs, ces travaux existants sont spécifiques à l'algorithme afin d'exploiter les caractéristiques spécifiques celui-ci dans le but de réduire la complexité arithmétique. En effet, Xu et al. exploitent la propriété de communication BPSK, i.e. les valeurs -1 et 1, pour simplifier les opérations arithmétiques dans leur réalisation d'un ASIC de l'algorithme PIC à multi-étages [114]. Étant donné que la différence des sorties de deux étages tend à zéro lors que l'annulation des interférences converge, les auteurs de [114] ont développé le MPIC différé. Dans le MPIC différé, au lieu d'opérer sur les sorties de chaque étage, il s'effectue sur la différence des sorties de deux étages consécutifs. Lorsque l'annulation converge, le nombre de zéros dans ce vecteur différé des sorties de deux étages augmente. Étant donné les données BPSK avec sorties dures, la valeur des non zéros de ce vecteur différé sont ± 2 qui sont le résultat de l'addition de deux valeurs de ± 1 [114]. Ainsi, la multiplication avec ces non zéros de ± 2 peuvent être simplement réalisée par un décalage. Ce MPIC présente les mêmes performances algorithmiques par rapport au MPIC standard. Aussi, Yuanbin et al. ont exploité la propriété des données binaires pour simplifier la version NLMS [116]. Ils peuvent donc éviter la division dans l'algorithme d'adaptation NLMS.

Rajagobal et al. ont proposé l'architecture VLSI parallèle d'un MPIC [90]. Ces derniers auteurs ont aussi modifié l'identification du canal à usagers multiples basée sur le maximum vraisemblance (*ML - Maximum likelihood*) par une méthode itérative pour éviter l'inversion des matrices dans l'implémentation VLSI [90]-[91]. Ces travaux consistent à décomposer le récepteur en plusieurs tâches implantables. Toutefois, la complexité d'implémentation de ces travaux est énorme. En fait, une implémentation pour 16 usagers à OVSF=16 et une longueur de données de 16 bits, la surface exige 10G transistors pour répondre en temps réel [90]. D'ailleurs, l'effort d'accélérer le temps d'exécution repose sur la méthode arithmétique en ligne afin d'accélérer le temps d'exécutions des opérateurs arithmétiques [31].

Ajoutons aussi que plusieurs implémentations en ASIC des récepteurs moins complexes ont également été proposées. En effet, [26], [100] ont implémenté des filtres MF. Pedersen et al. ont implémenté un SIC [85]. Toutefois, ces travaux ont répondu aux algorithmes peu sophistiqués – MF [26], [100], *Rake* [36], SIC [85] – qui offrent une faible performance pour les systèmes CDMA en présence des MAI par rapport aux autres récepteurs standards MMSE et PIC [20].

Une autre direction de recherche concerne des travaux de l'architecture avancée des processeurs qui sont aptes aux applications des récepteurs sophistiqués incluant la MUD. En effet, pour assurer à la fois les performances des algorithmes et le temps d'exécution en temps réel, le processeur doit avoir une structure super-scalaire particulière, comme par exemple le processeur à flux des données⁶ *Imagine* [89]. Le processeur *Imagine* consiste en 8 grappes (*cluster*) d'UAL (Unité Arithmétique et Logique), une file de registres à flux de données, un système de mémoires à flux de données, ainsi qu'un contrôleur arbitrant le flux de données entre des grappes d'UAL et la file de registres. Le système de mémoires dispose d'un mécanisme d'ordonnancement permettant d'exploiter le débit maximum de 4 SRAM externes. L'architecture particulière avec hiérarchie de flux des données permet de maximiser les performances crêtes du processeur. Un tel processeur pourrait être exploité pour l'application MUD complexe [89]. Une autre architecture, nommé *picoArray*TM de *picoChip* [86] serait d'intérêt pour une application extensive de calcul comme la MUD. Par exemple, un processeur PC203 de *picoChip* [86] est doté des tableaux des éléments de traitement DSP qui est dans le fond un MAC. Ce processeur est également intégré des coprocesseurs réalisant les fonctions comme FFT/IFFT, FEC (Turbo, Reed-Solomon et Viterbi) et cryptographie intégrée (AES/DES/3DES).

À notre avis, la limitation de ces travaux proposés existants se trouve encore dans la gestion de mémoires exigée par les algorithmes complexes de MUD et plus particulièrement quand le nombre d'usagers à traiter augmente. Il est important de noter que les propositions d'implémentation en technologie VLSI ne démontrent pas la capacité de maximiser le nombre d'usagers détectés pour un niveau de complexité donné. Cet inconnu a fait partie de nos objectifs.

⁶ *Stream processor*

Tenant compte des limitations des outils commerciaux à l'heure actuelle, un autre aspect de la problématique concerne la méthodologie de conception. En fait, à notre avis, il n'existe pas encore un flot de conception transparent et systématique. En fait, les flots de conception commerciaux [68], [112]-[113] et institutionnels [12]-[13], [22], [41] qui permettent le développement rapide existent. Aussi, il existe des outils capables de générer une architecture matérielle complexe [12]-[13], [112]-[113]. Cependant, ils sont généraux pour réaliser une application spécifique exigeant une structure et un contrôle particulier. Le contrôle complexe consiste à arbitrer le fonctionnement de l'architecture même – pipeline, ordonnancement, gestion de mémoires, d'accès de données et d'adresse – et aussi l'interface avec l'extérieur.

Une bonne méthodologie doit, à notre avis, être capable d'estimer les ressources exigées par un algorithme juste à partir de sa spécification algorithmique. En plus, il est nécessaire d'impliquer les langages de haut niveau dans des développements matériels des systèmes complexes afin d'exploiter l'interopérabilité facile de ces derniers. Cela nous a menés à une méthodologie de conception matérielle en utilisant les langages C/C++, MATLAB® et SystemC.

Le tableau 3 résume les travaux d'implémentation des récepteurs à usagers multiples en ordre alphabétique des noms des premiers auteurs. Il s'agit d'une comparaison relative des travaux existants les plus significatifs. Dans ce tableau, on entend par « Performance algorithmique » les performances en termes de taux d'erreur BER des récepteurs en question. On entend par « Complexité matérielle » la quantité de ressources matérielles nécessaire pour une implémentation FPGA ou DSP et la surface pour une implémentation ASIC.

Dans la colonne « Complexité matérielle », on attribue « non-définie » si les auteurs ne mentionnent pas de complexité matérielle. Aussi, dans la colonne « Norme », on attribue « Non-définie » si les auteurs ne mentionnent pas de norme CDMA qu'ils ciblent dans leurs travaux. Dans la colonne « Maximiser le nombre d'usagers sur un dé silicium », on attribue « Non » si le travail respectif ne maximise pas le nombre d'usagers qu'il peut traiter sur une seule puce. Parmi ces travaux, on n'en trouve que deux travaux (Xu et al. [114] et TI [54]) visent la norme WCDMA. Cependant, le travail de TI est basé sur le *Rake* que l'on connaît ses inconvénients en performances algorithmiques. Le travail de Xu et al. vise un système avec un nombre limité

d'usagers (8 en effect) et l'architecture n'est pas reconfigurable étant donné sa cible ASIC. Le travail de Rajagobal et al. [90] répond bien au niveau architectural.

Tableau 3 : Résumé des travaux d'implémentation des récepteurs à usagers multiples.

No.	Travaux	Année	Récepteur MUD	Technologie cible	Performance algorithmique	Norme	Complexité matérielle	Maximiser le nombre d'usagers sur un dé silicium
1	Ching-Hung et al. [26]	2001	MF	ASIC	Très faible	Non-définie	Faible	Non
2	Correal et al. [28]	1999	PIC à 2 étages	DSP	Élevé	Non-définie	Non-définie	Non
3	Eltawil et al. [36]	2004	<i>Rake</i> à 2 <i>fingers</i>	ASIC	Faible	Non-définie	Faible	Non
4	TI [54]	2000	<i>Rake</i> à 6 <i>fingers</i>	DSP	Faible	WCDMA	Non-définie	Non
5	Lequepeys et al. [62]	1998	MPIC à 2 étages	ASIC et prototypé sur FPGA	Élevé	Non-définie	Non-définie	Non
6	Leung et al. [63]	2000	<i>Rake</i> à 12 <i>fingers</i>	FPGA	Faible	IS-95	Moyenne	Non
7	Xu et al. [114]	2002	MPIC à 4 étages	ASIC	Élevé	WCDMA	Moyenne	Non
8	Rajagobal et al. [90]	2002	MPIC à 3 étages	ASIC	Élevé	Non-définie	Très élevé	Non

En fait, Rajagobal et al. ont bien exploité les techniques avancées de parallélisme en partitionnant l'algorithme complexe d'un MPIC en plusieurs tâches plus intégrables. Cependant, pour répondre à un système avec un nombre d'usagers suffisamment important à un débit suffisamment élevé, le parallélisme maximal de l'architecture développée introduit un surplus significatif comme indiqué ci-haut. Toutefois, ces deux travaux (Xu et al. [114] et TI [54])

répondent à certains de nos critères concernant les performances algorithmiques en taux d'erreur binaire et performances en temps réel de calcul.

II.5. Librairies arithmétiques en virgule fixe

Tenant compte de l'importance de l'arithmétique à virgule fixe en conception, plusieurs travaux relatifs ont été proposés. Nous allons citer quelques travaux représentatifs et pertinents à nos travaux. Il s'agit des librairies en virgule fixe.

II.5.1. Proposition de IEEE

Ce paquetage a été proposé en 2003 afin de standardiser la mathématique à virgule fixe en VHDL [15]. Ce paquetage se base sur le paquetage standard d'arithmétique IEEE *numeric_std*. Cette proposition introduit deux types de données fixes *sfixed* et *ufixed*. Ces deux derniers sont pour la virgule fixe signée et non-signée respectivement. Un nombre fixe de ce paquetage est alors défini comme un vecteur de type *numeric*. Contrairement à la syntaxe de base du langage VHDL qui considère l'index des tableaux comme un nombre naturel non-négatif, ce paquetage introduit l'index négative.

Un nombre fixe est alors défini, par cette proposition, comme suit :

$$\begin{aligned} & \text{sfixed}(I \text{ downto } -F) ; \text{ -- pour un nombre signé} \\ & \text{ufixed}(I \text{ downto } -F) ; \text{ -- pour un nombre non-signé} \end{aligned}$$

où $I+1$ et F dénotent respectivement le nombre de bits entiers et fractionnaires.

Un nombre fixe peut être représenté comme un *little andian* (comme ci-haut) ou un *big endian* comme suit :

$$\begin{aligned} & \text{sfixed}(-F \text{ to } I) ; \text{ -- pour un nombre signé} \\ & \text{ufixed}(-F \text{ to } I) ; \text{ -- pour un nombre non-signé} \end{aligned}$$

II.5.2. Toolbox de MATLAB®

Le *toolbox* à virgule fixe MATLAB® a été développé pour l'arithmétique à virgule fixe dans l'environnement de MATLAB® [65]. Il se base sur un objet MATLAB® nommé *fi*. Un objet a trois propriétés constitutantes : *données*, *fimath* et *numerictype*.

- Propriété données constitue des données de l'objet *fi*, et est toujours inscriptible.
- Propriété *fimath*, qui est un objet en soi, est automatiquement créée quand un objet *fi* est créé. Celle-ci définit les propriétés de l'arithmétique à virgule fixe relative à l'objet tels que la longueur de bits des résultats intermédiaires, le mode de produit, le mode de l'addition, etc.
- Propriété *numerictype*, qui est également un objet en soi, est automatiquement créée quand un *fi* objet est créé. Celle-ci définit les propriétés d'un nombre fixe tels que le nombre de bits, nombre de bits entiers, nombre de bits fractionnaires, mode de débordement, etc.

Une fois qu'une variable est définie comme un objet *fi*, des opérations arithmétiques s'appliquent sur cette variable d'objet comme des opérations arithmétiques ordinaires.

II.5.3. Bibliothèques arithmétiques de SystemC

Les bibliothèques virgule fixe de SystemC ont été développées pour modéliser des modèles SystemC virgule fixe [83]. En SystemC, un nombre à virgule fixe binaire se représente par une séquence de bits avec une position spécifiée pour le point binaire. Les bits à gauche du point binaire représentent la partie entière, alors que les bits à droite du point binaire représentent la partie fractionnaire. Une variable SystemC à virgule fixe est alors caractérisé par les éléments ci-dessous :

- La longueur (*wl*) qui est le nombre de bits total dans la représentation d'un nombre fixe.

- La longueur de la partie entier (iwl) qui est le nombre de bits entiers représentant la partie entière.
- Le codage des bits : signé/non-signé, complément à deux.

Le bit à l'extrême gauche dénote le bit le plus significatif (MSB), alors que le bit à l'extrême droite dénote le bit le moins significatif (LSB). Dans le cas où le nombre fixe est signé, le MSB est aussi le bit de signe. Dans la représentation SystemC, le point binaire peut être à l'extérieur des bits de données, i.e. ceci peut être à gauche du MSB ou à droite du LSB d'un certain nombre de bits. Ainsi, il y a trois cas de représentation d'un nombre fixe selon la position du point binaire comme suit :

- Le point binaire est à droite du bit LSB ($wl < iwl$). Dans ce cas, il y a $(iwl - wl)$ zéros entre le LSB et le point binaire, par exemple : $\times\times\times\times00\bullet$ ($wl = 5, iwl = 7$).
- Le point binaire est à gauche du bit MSB ($iwl < 0$). Dans ce cas, il y a $(|iwl|)$ bits de signe entre le MSB et le point binaire, par exemple : $\bullet ss\times\times\times\times$ ($wl = 5, iwl = -2$).
- Le point binaire est au milieu du bit MSB et LSB ($0 \leq iwl \leq wl$). C'est le cas ordinaire d'un nombre fixe, par exemple : $\times\times\bullet\times\times\times\times$ ($wl = 7, iwl = 2$).

II.5.4. Types de données de Mentor Graphics : Algorithmic C™

Les types de données Algorithmic C™ [70] ont été développés par la société Mentor Graphics afin de promouvoir leur outil Catapult® C [71]. Ce dernier est un outil de synthèse de haut niveau qui est capable de synthétiser des modèles en C/C++ standards. Le développement de ces types de données est basé sur les techniques dites des *templates*. Ces dernières imposent que les longueurs des nombres virgule fixe soient statiques.

II.6. Conclusions

Ce chapitre a résumé les bases sur la communication DS-CDMA ainsi que les récepteurs à usagers multiples les plus représentatifs dans la littérature. Notamment, le travail de ce chapitre fait une synthèse des travaux pertinents concernant l'implémentation VLSI des récepteurs à usagers multiples complexes, la détection à usagers multiples multi-débit et les méthodologies de conception. Les travaux d'implémentation VLSI des récepteurs sophistiqués de manière à répondre en temps réel aux scénarios de communication restent encore non-triviaux. Plusieurs efforts ont été effectués. Toutefois, les travaux pertinents sont spécifiques à l'algorithme qui exploitent principalement la particularité des systèmes visés pour simplifier le calcul arithmétique. D'ailleurs, ces travaux ne présentent pas un nombre d'usagers sur un seul composant. Une autre limitation importante des travaux d'implémentation VLSI concerne aussi la méthodologie de conception. Ces limitations attirent ainsi notre attention. Concernant la détection multi-débit, les méthodes couramment utilisées dans la littérature ont été décrites. Parmi celles-ci, les méthodes LRD et HRD sont adoptés pour nos travaux dans le chapitre IV. Des travaux dans la littérature ont montré que la méthode LRD offre des performances meilleures que la méthode HRD. Toutefois, il introduit un surplus énorme de surface.

CHAPITRE III

ARCHITECTURES PROPOSÉES DU RÉCEPTEUR CF-MUD

Ce chapitre présente l'architecture matérielle du CF-MUD proposée ciblée sur FPGA. Les composants FPGA ciblés sont les familles Virtex-II et Virtex-II Pro de Xilinx[®]. Il est à noter que le développement de ces architectures prend en compte de la particularité de la technologie ciblée. Ainsi, ces architectures sont spécifiques pour ces FPGA cibles. Toutefois, l'architecture générale est également applicable aux autres technologies grâce à sa régularité. La description succincte du récepteur CF-MUD aidera la compréhension de cet algorithme. Plus de détails sur le CF-MUD se trouvent dans [30].

III.1. Description de l'algorithme CF-MUD

La figure 22 présente la structure générale de l'algorithme de détection à cascade des filtres adaptatifs (*CF-MUD : Cascade filter MUD*). Le cœur du récepteur CF-MUD comprend deux blocs de filtrage : le bloc Signature et le bloc Détection. Le traitement de la rétroaction des données est optionnel, et permet de renforcer les performances en tenant compte des données estimées pour détecter les données ultérieures. Toutefois, la rétroaction des données introduit une forte dépendance aux données, laquelle complique les mécanismes de parallélisme en implémentation matérielle. Dans la structure générale du CF-MUD, il peut aussi avoir un réseau d'antennes multiples et un sur-échantillonnage à l'entrée [30].

Dans le cadre de ce travail de thèse, on considère la structure sans rétroaction des données du CF-MUD. Il se compose alors de deux blocs en cascade de filtres adaptatifs – le bloc Signature et le bloc Détection – et du bloc de génération des séquences d’entraînement interne. L’adaptation consiste à déterminer les coefficients des filtres transversaux (*feed-forward*).

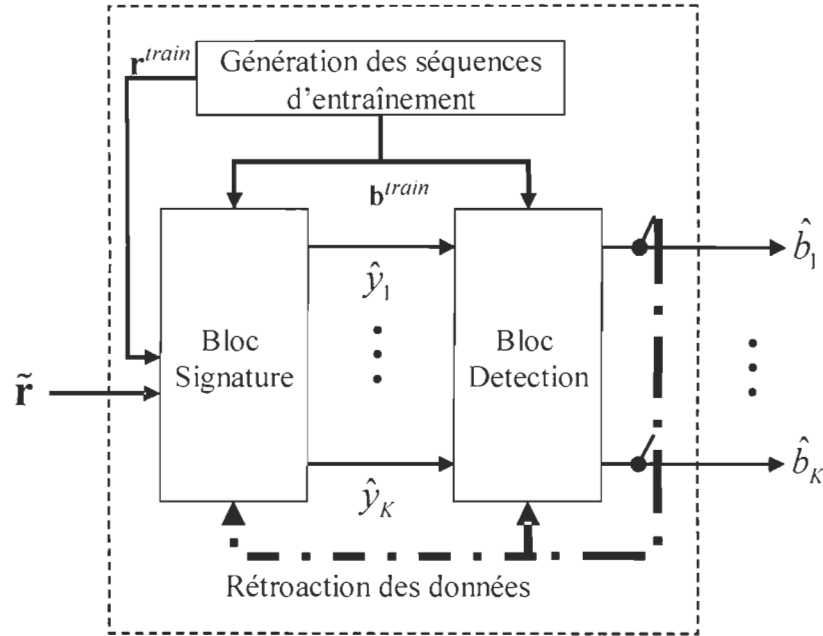


Figure 22 : Principe du récepteur CF-MUD. La rétroaction des données est optionnelle.

Considérant la non-disponibilité dans les normes (3GPP et 3GPP2) des données transmises par les usagers servant à l’apprentissage pour l’adaptation des coefficients, un processus de séquences de données virtuelles d’apprentissage est réalisé à l’intérieur du détecteur à la réception. Il est à noter que les seules données transmises par les transmetteurs mobiles sont les données de contrôle ou plus spécifiquement les données pilotes. Le nombre de ces données est insuffisant pour assurer la convergence et le suivi des canaux variants des mobiles. C’est pourquoi une technique particulière de génération des séquences d’entraînement a été proposée [30]. Elle consiste simplement à utiliser le modèle des canaux selon l’équation (4) dont les coefficients ($h_{n,l}$) et les délais ($\tau_{n,l}$) sont obtenus à partir d’une méthode d’identification des canaux. La méthode communément utilisée est le corrélateur [110].

Se basant sur les séquences des données d'apprentissage, plusieurs techniques adaptatives sont disponibles pour effectuer l'adaptation [47] : moyennes des moindres carrés (*LMS : Least mean square*), LMS normalisé (*NLMS : Normalized LMS*), moindres carrés récurrents (*RLS : Recursive least square*), Kalman, etc. Dans ce projet doctoral, nous considérons la technique des moindres carrés LMS.

III.1.1. Bloc Signature

Le principe du bloc Signature est présenté à la figure 23. Cette partie du récepteur permet d'adapter les Signatures⁷ des usagers sans connaître au préalable leurs codes pseudo-aléatoires. Cette partie du récepteur consiste en K Signatures où chacune est attribuée à un usager. Pour les codes courts, c.-à-d. de longueur 256, chaque Signature consiste en $256/N_c$ filtres adaptatifs. Le bloc Signature en soi se compose de deux phases : la phase d'adaptation et la phase de détection.

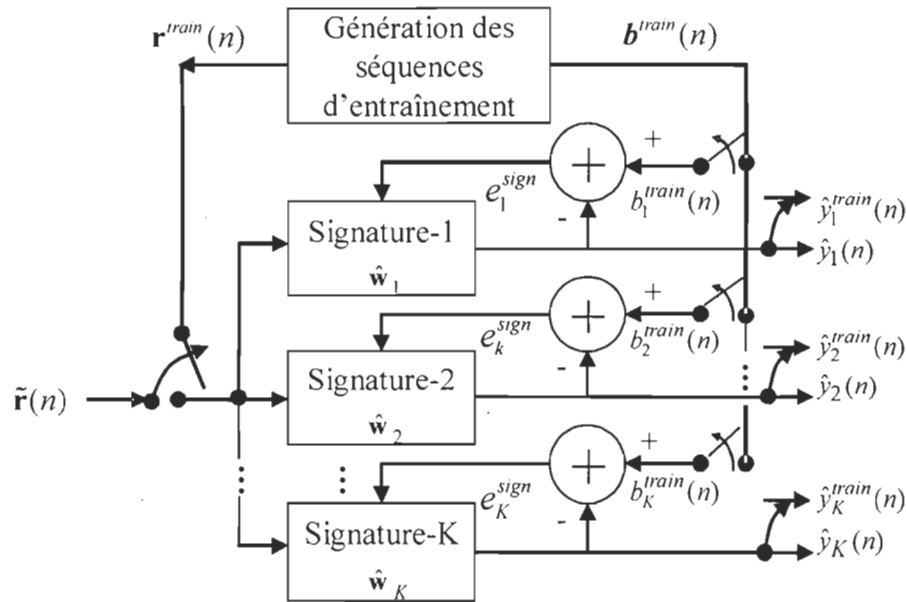


Figure 23 : Principe de la cascade des filtres adaptatifs du bloc Signature.

La phase d'adaptation consiste à adapter les poids des filtres adaptatifs $\hat{\mathbf{w}}_k(n)$ de chaque usager k , et s'exprime comme suit :

⁷ Sauf indication contraire, Signature (avec S majuscule) signifie un composant du bloc Signature, alors que signature (avec s minuscule) signifie le code signature ou code d'étalement qui est spécifique à chaque usager.

$$\hat{y}_k^{train}(n) = \hat{\mathbf{w}}_k^H(n) \mathbf{r}_k^{train}(n) \quad (38)$$

$$e_k^{SIGN}(n) = b_k(n) - \hat{y}_k^{train}(n) \quad (39)$$

$$\hat{\mathbf{w}}_k(n+1) = \hat{\mathbf{w}}_k(n) + \mu^{SIGN} \mathbf{r}_k^{train}(n) e_k^{SIGN}(n)^*, \text{ avec } \dim(\hat{\mathbf{w}}_k(n)) = N_f \times 1 \quad (40)$$

où : $\hat{\mathbf{w}}_k(n)$ dénote le vecteur du filtre adaptatif de l'utilisateur k ;

N_f dénote la longueur de ces filtres, avec $N_f = 2N_c$;

$b_k(n)$ est la séquence des symboles d'apprentissage, qui est intérieurement générée dans le récepteur pour l'adaptation de l'utilisateur k ;

$*$ dénote la valeur complexe conjuguée ;

$e_k^{SIGN}(n)$ dénote l'erreur d'adaptation de l'utilisateur k du bloc Signature ;

$\mathbf{r}_k^{train}(n)$ dénote le vecteur d'entrée des données, qui est intérieurement généré dans le récepteur en utilisant $b_k(n)$ et les estimés des canaux, pour l'adaptation de l'utilisateur k ;

μ^{SIGN} dénote le pas d'apprentissage du bloc Signature.

Les vecteurs $\hat{\mathbf{w}}_k(n)$ et $\mathbf{r}_k^{train}(n)$ sont respectivement définis comme suit :

$$\hat{\mathbf{w}}_k(n) = [\hat{w}_k(nT + T_c), \hat{w}_k(nT + 2T_c), \dots, \hat{w}_k(nT + N_f T_c)]^T \in \mathbb{C}^{N_f \times 1} \quad (41)$$

$$\mathbf{r}_k^{train}(n) = [r_k^{train}(nT + T_c + \tau_k), r_k^{train}(nT + 2T_c + \tau_k), \dots, r_k^{train}(nT + N_f T_c + \tau_k)]^T \in \mathbb{C}^{N_f \times 1} \quad (42)$$

où τ_k dénote le délai relatif de l'utilisateur k .

Lorsque la phase d'adaptation se termine, ces vecteurs de poids estimés sont utilisés par la phase de détection afin de détecter les symboles des usagers. La phase de détection s'exprime comme suit :

$$\hat{y}_k(n) = \hat{\mathbf{w}}_k^H(n) \tilde{\mathbf{r}}_k(n) \quad (43)$$

où $\tilde{\mathbf{r}}_k(n)$ est le vecteur d'entrée des données bruitées à estimer de l'utilisateur k , et s'exprime comme suit :

$$\tilde{\mathbf{r}}_k(n) = [\tilde{r}(nT + T_c + \tau_k), \tilde{r}(nT + 2T_c + \tau_k), \dots, \tilde{r}(nT + N_f T_c + \tau_k)]^T \in \mathbb{C}^{N_f \times 1} \quad (44)$$

III.1.2. Bloc Détection

Le principe du bloc Détection est présenté à la figure 24. Le but du bloc Détection consiste à éliminer les MAI résiduels après le bloc Signature. Il effectue donc l'annulation des interférences à partir des signaux estimés de tous les usagers du bloc Signature. Cette partie du récepteur consiste en K filtres adaptatifs où chacune est attribuée à un usager. Comme le bloc Signature, il se compose aussi de deux phases : la phase d'adaptation et la phase de détection. Alors que la phase d'adaptation vise à ajuster les paramètres des filtres des usagers, la phase de détection vise à détecter les symboles des usagers tout en utilisant les poids adaptés par ces derniers.

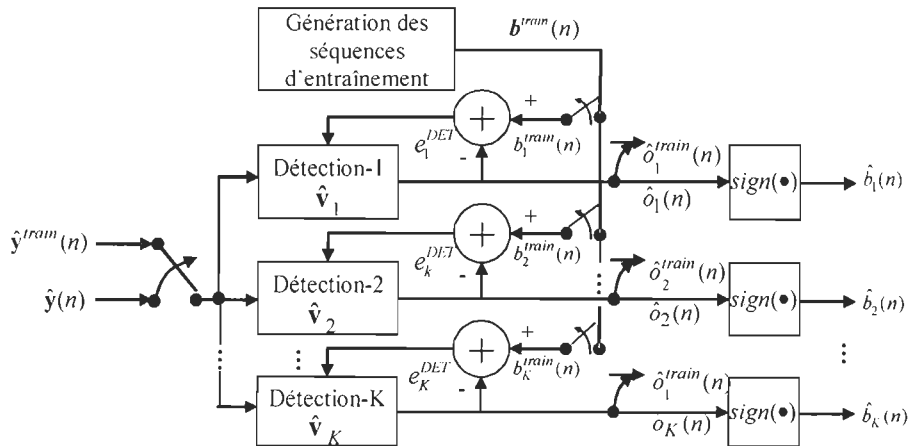


Figure 24 : Principe de la cascade des filtres adaptatifs du bloc Détection.

Tous les filtres Détection utilisent le même vecteur d'entrée des données, qui est formé par les sorties du bloc Signature, pour tous les usagers, mais avec les délais respectifs différents. La phase d'adaptation du bloc Détection s'exprime comme suit :

$$\hat{o}_k^{train}(n) = \hat{\mathbf{v}}_k(n)^H \hat{\mathbf{y}}^{train}(n) \quad (45)$$

$$e_k^{DET}(n) = b_k(n) - \hat{o}_k(n) \quad (46)$$

$$\hat{\mathbf{v}}_k(n+1) = \hat{\mathbf{v}}_k(n) + \mu^{DET} e_k^{DET}(n) \hat{\mathbf{y}}^{train}(n) \quad (47)$$

où $\hat{\mathbf{v}}_k(n)$, avec $\dim(\hat{\mathbf{v}}_k(n)) = 3K \times 1$, dénote le vecteur des poids du filtre adaptatif de l'utilisateur k ;

$b_k(n)$ est la séquence des symboles d'apprentissage, qui est la même que celle du bloc Signature, de l'utilisateur k ;

$\hat{\mathbf{y}}^{train}(n)$, qui est le même vecteur d'entrée pour tous les filtres adaptatifs, dénote les données pour l'adaptation ;

$e_k^{DET}(n)$ dénote l'erreur d'adaptation de l'utilisateur k du bloc Signature ;

μ^{DET} dénote le pas d'apprentissage du bloc Détection.

Les vecteurs $\hat{\mathbf{v}}_k(n)$ et $\hat{\mathbf{y}}^{train}(n)$ sont respectivement définis comme suit :

$$\hat{\mathbf{v}}_k(n) = [\hat{v}_k(nT+1), \dots, \hat{v}_k(nT+KT_c), \dots, \hat{v}_k(nT+2KT), \dots, \hat{v}_k(nT+2KT)]^T \in \mathbb{C}^{3K \times 1} \quad (48)$$

$$\hat{\mathbf{y}}^{train}(n) = [\hat{y}_1^{train}(n-1), \dots, \hat{y}_K^{train}(n-1), \dots, \hat{y}_K^{train}(n), \dots, \hat{y}_K^{train}(n+1)]^T \in \mathbb{C}^{3K \times 1} \quad (49)$$

La phase d'adaptation du bloc Détection s'exprime comme suit :

$$\hat{o}_k(n) = \hat{\mathbf{v}}_k(n)^H \hat{\mathbf{y}}(n) \quad (50)$$

$$\hat{b}_k(n) = \text{sign}(\hat{o}_k(n)) \quad (51)$$

où $\hat{o}_k(n)$ et $\hat{b}_k(n)$ dénotent respectivement la sortie douce et la sortie dure de l'utilisateur k du bloc Détection.

III.2. Architecture générale

La structure générale du récepteur CF-MUD est présentée à la figure 25. Elle consiste en une organisation interne – partie désignée par ② – et une structure de mémoires externes. L'organisation interne est à l'intérieur du FPGA cible, alors que les mémoires externes sont à l'extérieur du FPGA cible. L'architecture matérielle se base sur les tableaux réguliers et modulaires à deux étages des unités de traitement, aussi appelées processeurs élémentaires (*PE – Processing element*). Dans la figure 25, ces derniers appartiennent au noyau ①. La structure de mémoires internes comprend les mémoires semi-globales et les mémoires globales. Le partage de ressources des PE se base sur les multiplexeurs. Les données sont de valeurs complexes, sauf les sorties qui sont de valeurs réelles⁸. Chaque PE est un processeur en soi avec le contrôle local. Le contrôle global sert à contrôler l'activité de l'architecture, la stratégie de pipeline, et la structure de mémoires internes, de même que l'interface avec l'extérieur.

Le noyau de l'architecture se compose d'une structure de mémoire tampon d'entrée *InputBuffer*, d'un tableau des PE ③, d'une structure de mémoire tampon intermédiaire *InterBuffer*, d'un tableau des PE ④, et d'une structure de mémoire tampon de sortie *OutputBuffer*.

Le noyau ① de l'architecture est présenté à la figure 26. Il s'agit du *mapping* de l'algorithme CF-MUD sur les tableaux des PE et les mémoires internes. Sauf indication

⁸ Comme on considère les données BPSK.

contraire, les données sont complexes. Dans la figure 26, les bus de données et les blocs mémoires sont simplifiés pour faciliter la lecture de cette figure. L'architecture est reconfigurable pour plusieurs scénarios de communications. Du coup, un scénario de communications est caractérisé par les paramètres algorithmiques : nombre d'itérations d'adaptation, pas d'adaptation, nombre d'utilisateurs actifs à traiter, le débit de données caractérisé par le facteur d'étalement OVFS. Lorsqu'un ou plusieurs de ces paramètres changent, l'architecture est capable de changer adéquatement. Par conséquent, dans notre conception architecturale, ces paramètres se traduisent en paramètres architecturaux de reconfiguration. Ainsi, ces paramètres de reconfiguration sont définis dans le tableau 4.

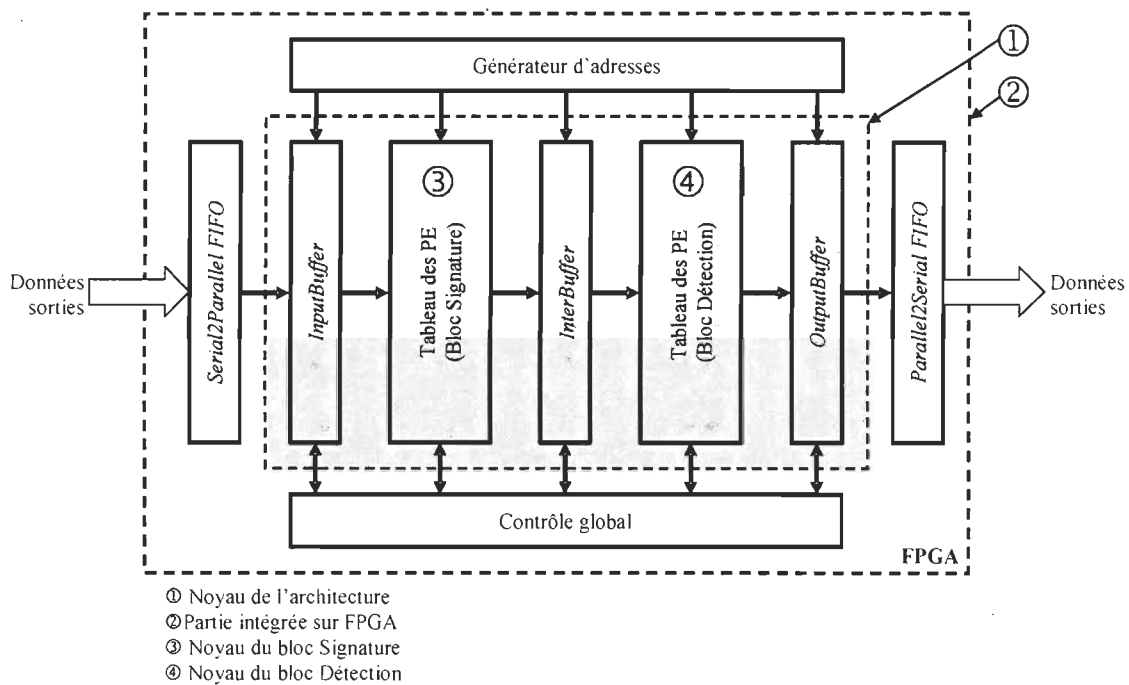


Figure 25 : Architecture matérielle générale du CF-MUD.

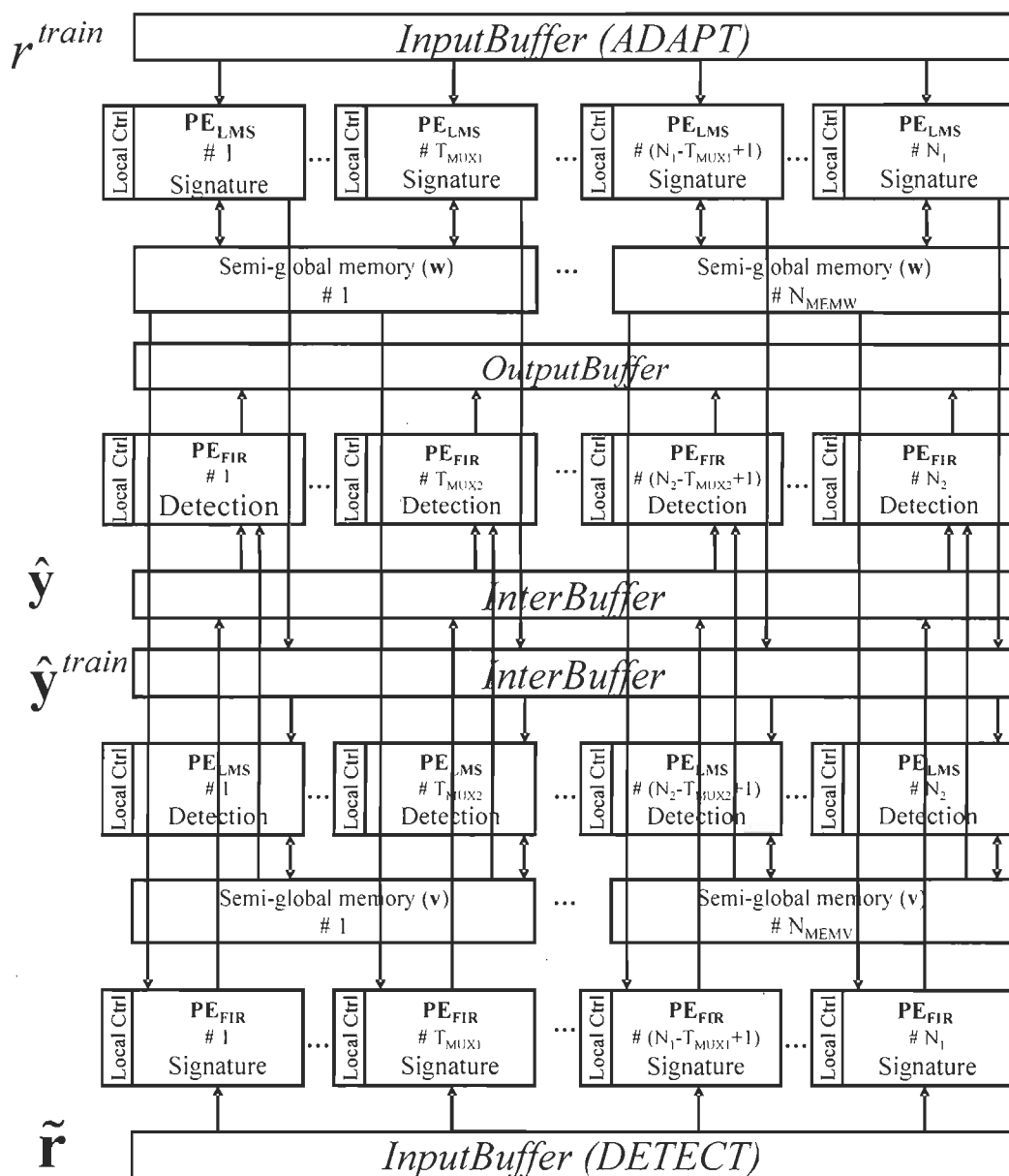


Figure 26 : Organisation des tableaux des PE, et des mémoires internes.

Détection. Les mêmes filtres adaptatifs FIR-LMS sont utilisés dans la phase d'adaptation aussi bien du bloc Signature et que du bloc Détection. Également, les mêmes filtres transversaux FIR sont utilisés dans la phase de détection aussi bien du bloc Signature et que du bloc Détection.

Tableau 4 : Paramètres de reconfiguration de l'architecture

Nom du paramètre	Description
T_{MUX1}	Multiplexage temporel des PE du bloc Signature
T_{MUX2}	Multiplexage temporel des PE du bloc Détection
N_1	Nombre de PE utilisés par le bloc Signature
N_2	Nombre des PE utilisés par le bloc Détection
N_{MEMW}	Nombre de mémoires semi-globales pour les vecteurs de coefficients $\hat{\mathbf{w}}$ des filtres du bloc Signature
N_{MEMV}	Nombre de mémoires semi-globales pour les vecteurs des coefficients $\hat{\mathbf{v}}$ des filtres du bloc Détection

III.3. Noyau du bloc Signature

La figure 27 présente le noyau du bloc Signature. Il s'agit du *mapping* du bloc Signature exprimé par les équations (38-40, 43) sur une structure régulière d'un tableau régulier des PE_{LMS} , d'un tableau régulier des PE_{FIR} , d'un tableau régulier des mémoires semi-globales, d'une mémoire globale *InputBuffer*, et d'une mémoire globale *InterBuffer*. La mémoire *InputBuffer* se compose de deux mémoires à données complexes dénotées par *InputBuffer(ADAPT)* et *InputBuffer(DETECT)*. Ces dernières sont utilisées pour temporiser respectivement les données d'entraînement (r^{train}) de la phase d'adaptation et les données à détecter (\tilde{r}) de la phase de détection. La mémoire *InterBuffer* se compose de deux mémoires à données complexes dénotées par *InterBuffer(ADAPT)* et *InterBuffer(DETECT)*. Ces dernières sont utilisées pour mémoriser les sorties estimées respectives de la phase d'adaptation et de celle de détection.

Chaque mémoire semi-globale se partage un nombre d'unités de traitement PE_{LMS} et PE_{FIR} . Ces mémoires semi-globales sont utilisées pour mémoriser les vecteurs des poids adaptés du bloc Signature.

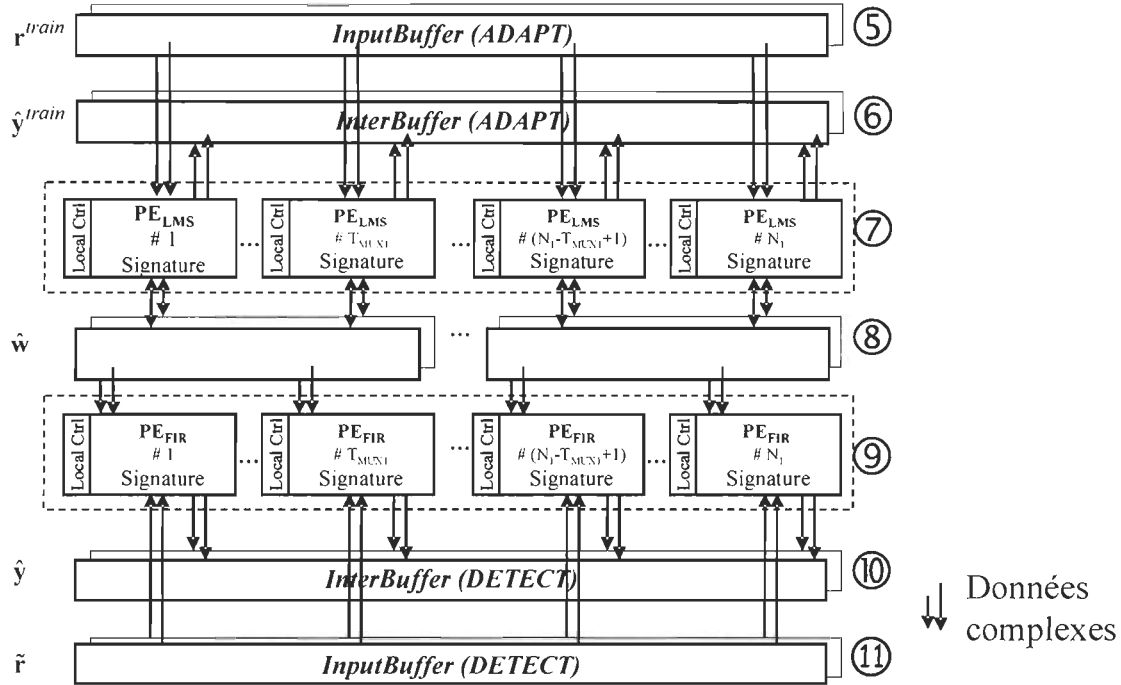


Figure 27 : Mapping du bloc Signature sur le tableau des PE et les mémoires internes. Les deux flèches parallèles signifient les données complexes.

III.4. Noyau du bloc Détection

La figure 28 présente le noyau du bloc Détection. Il s'agit du *mapping* du bloc Détection – exprimé par les équations (45-47, 50) – sur une structure régulière d'un tableau régulier des PE_{LMS} , d'un tableau régulier des PE_{FIR} , d'un tableau régulier des mémoires semi-globales, d'une mémoire globale *OutputBuffer*, et d'une mémoire globale *InterBuffer*. La mémoire *InterBuffer* est la même que celle du noyau Signature. Chaque mémoire semi-globale se partage un nombre d'unités de traitement PE_{LMS} et PE_{FIR} . Ces mémoires semi-globales sont utilisées pour mémoriser les vecteurs des poids adaptés du bloc Détection. La mémoire globale

OutputBuffer est constituée de données réelles car on ne veut récupérer que la partie réelle des données estimées. Ces données réelles correspondent aux données BPSK transmises.

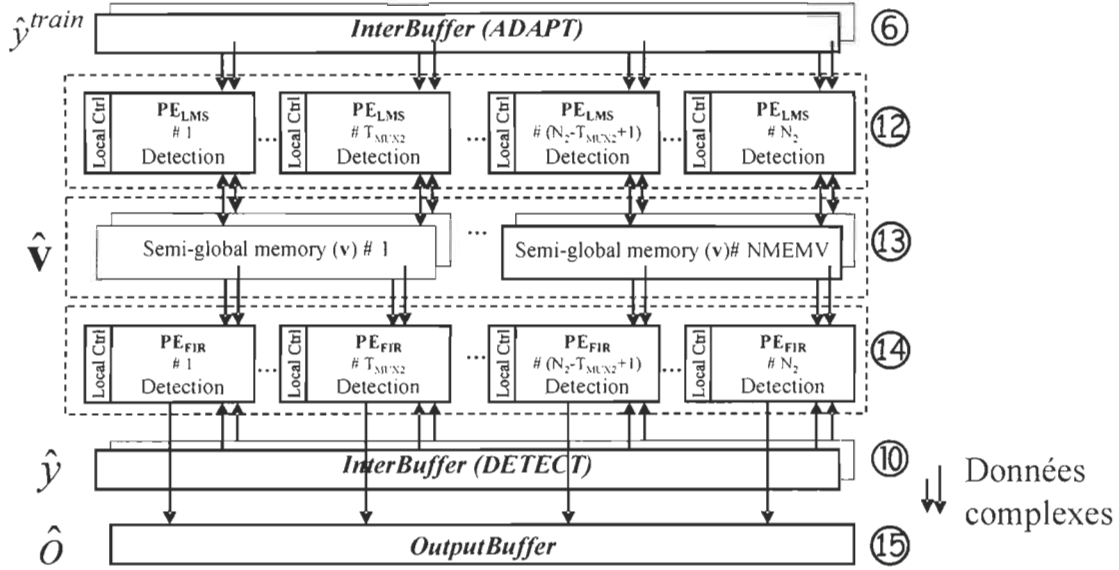


Figure 28 : Mapping du bloc Détection sur le tableau des PE et des mémoires internes. Les deux flèches parallèles signifient les données complexes.

III.5. Multiplexage temporel

Pour augmenter le taux d'utilisation de ces PE, et pour diminuer alors le temps d'exécution, chaque PE est utilisé pour traiter plusieurs usagers par multiplexage temporel. En effet, un seul PE est utilisé pour traiter un nombre d'usagers par sélection à multiplexage temporel. Pour le bloc Signature, nous utilisons le même multiplexage temporel – dénoté par T_{MUX1} . Ce multiplexage temporel est alors le même pour la phase d'adaptation et la phase de détection. Pour le bloc Détection, nous utilisons le même multiplexage temporel – dénoté par T_{MUX2} . Ce multiplexage temporel est le même pour la phase d'adaptation et la phase de détection.

III.6. Architecture des mémoires

Les trajets de données et d'adresses sont indépendants pour permettre l'accès simultané aux données et adresses. L'architecture de mémoires consiste en une structure des mémoires internes et en une structure des mémoires externes.

La structure des mémoires internes est intégrée à l'intérieur des FPGA cibles. Elle est connectée directement au noyau de l'architecture matérielle du récepteur CF-MUD afin d'accélérer l'accès aux données.

La structure des mémoires internes se compose de :

- Deux mémoires tampons d'entrée : *InputBuffer(ADAPT)* et *InputBuffer(DETECT)*,
- Deux mémoires tampons intermédiaires : *InterBuffer(ADAPT)* et *InterBuffer(DETECT)*,
- D'une structure de mémoires semi-globales ⑧ pour mémoriser les vecteurs des poids des filtres du bloc Signature,
- D'une structure de mémoires semi-globales ⑬ pour mémoriser les vecteurs des poids des filtres du bloc Signature,
- D'une mémoire globale de sortie pour temporiser les sorties estimées du récepteur *OutputBuffer*.

La structure des mémoires externes, composées des SDRAM, est à l'extérieur des FPGA cibles.

Chaque mémoire tampon d'entrée – *InputBuffer(ADAPT)* et *InputBuffer(DETECT)* – se compose de deux blocs mémoires globales de longueur $2N_f + D_{\max}$, où N_f est la longueur des

filtres et D_{\max} est le délai maximal relatif entre usagers. Le nombre de blocs mémoires de chaque mémoire tampon est deux parce que les données sont complexes.

Chaque mémoire tampon intermédiaire – $InterBuffer(ADAPT)$ et $InterBuffer(DETECT)$ se compose de deux blocs mémoires de longueur $4K$, où K est le nombre d'usagers. Le nombre de blocs mémoires de chaque mémoire tampon intermédiaire est deux parce que les données sont complexes.

La notion de mémoire semi-globale signifie qu'un nombre d'unités de traitement (PE_{LMS} et PE_{FIR}) partagent une même mémoire. La notion de mémoire globale signifie que toutes les unités de traitement d'un tableau régulier partagent une seule mémoire.

Les mémoires tampons d'entrée/sortie $InputBuffer(ADAPT)$, $InputBuffer(DETECT)$ et $OutputBuffer$ sont à multiports pour maximiser le parallélisme d'accès des PE. La mémoire tampon d'entrée ($InputBuffer(ADAPT)$ et $InputBuffer(DETECT)$) contient les données nécessaires aux entrées des filtres des usagers : r^{train} pour la phase d'adaptation et \tilde{r} pour la phase de détection. Les délais relatifs entre usagers sont utilisés pour initialiser leurs adresses accédant à $InputBuffer(ADAPT)$ et $InputBuffer(DETECT)$. La méthode d'association directe est appliquée pour résoudre le problème d'échec d'accès mémoire tampon d'entrée $InputBuffer$. La mémoire tampon intermédiaire est utilisée pour mémoriser les données temporaires qui sont sorties du bloc Signature et qui sont entrées au bloc Détection.

Le nombre total de mémoires semi-globales est $2N_1 + 2N_2$ où $2N_1$ et $2N_2$ est respectivement le nombre de mémoires semi-globales utilisées dans le bloc Signature, et celui dans le bloc Détection. Le multiple de 2 est introduit parce que les données sont complexes [1 pour la partie réelle et 1 pour la partie imaginaire].

Chaque élément mémoire est une mémoire dual-port comme présentée dans la figure 29. Il se compose de deux canaux d'accès simultané à une zone de données commune. Chaque canal indépendant se compose d'une entrée de donnée, d'une sortie de donnée et d'un signal de

contrôle lecture/écriture. Les deux canaux se synchronisent sur la même horloge. Cet élément de mémoire a été conçu afin de faciliter l'accès simultané et parallèle des PE (PE_{LMS} et PE_{FIR}). Comme la zone de données est commune à deux canaux d'accès, le contrôle de mémoire gère les adresses d'accès afin d'éviter des échecs éventuels. Aussi, cet élément de mémoire a été conçu afin d'exploiter les blocs mémoires dédiées disponibles sur les dés de siliciums des FPGA cibles.

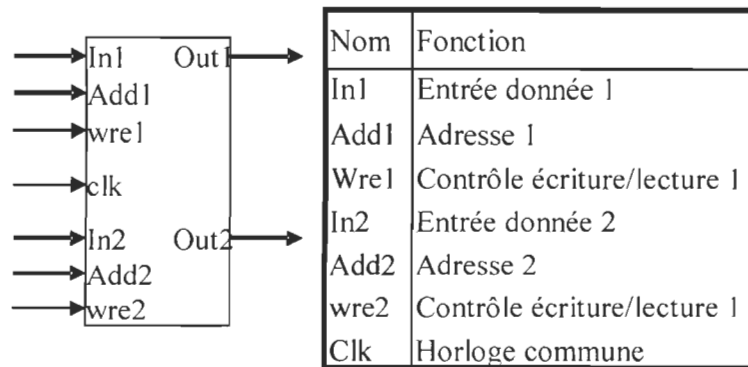


Figure 29 Mémoire dual-port.

La figure 30 présente la structure partielle d'une mémoire semi-globale des poids adaptés des filtres. La phase d'adaptation et de détection partage la même mémoire dual-port afin de faciliter leur accès simultané. Dans cette figure, WR/RD signifie que la lecture et l'écriture sont actives dans le canal de mémoire, alors que RD signifie que seule la lecture est active dans le canal. Évidemment, la phase d'adaptation (*ADAPT*) exige la lecture et l'écriture de mémoire, alors que la phase de détection (*DETECT*) n'exige que l'écriture. Pour simplifier la lecture de la figure, la figure 30 ne présente que la structure partielle de l'architecture d'un élément de mémoire semi-globale. En effet, l'exécution de la phase d'adaptation et de la phase de détection est respectivement effectuée par les PE_{LMS} et les PE_{FIR} .

La figure 31 présente le principe de la commutation d'accès mémoire de la phase d'adaptation et de détection. Ce principe d'accès aux mémoires semi-globales des poids adaptés des filtres est commun pour le bloc Signature et le bloc Détection. La flèche pale dénonce l'inactivité de l'accès aux mémoires, alors que la flèche noire foncée dénonce l'accès actif aux mémoires. Le flot d'états de cette commutation consiste en 3 états

$\{STATE0, STATE1, STATE2\}$. Dans l'état $STATE0$, seule la phase d'adaptation est active. Les deux phases – d'adaptation et de détection – sont actives dans l'état $STATE1$ et $STATE2$.

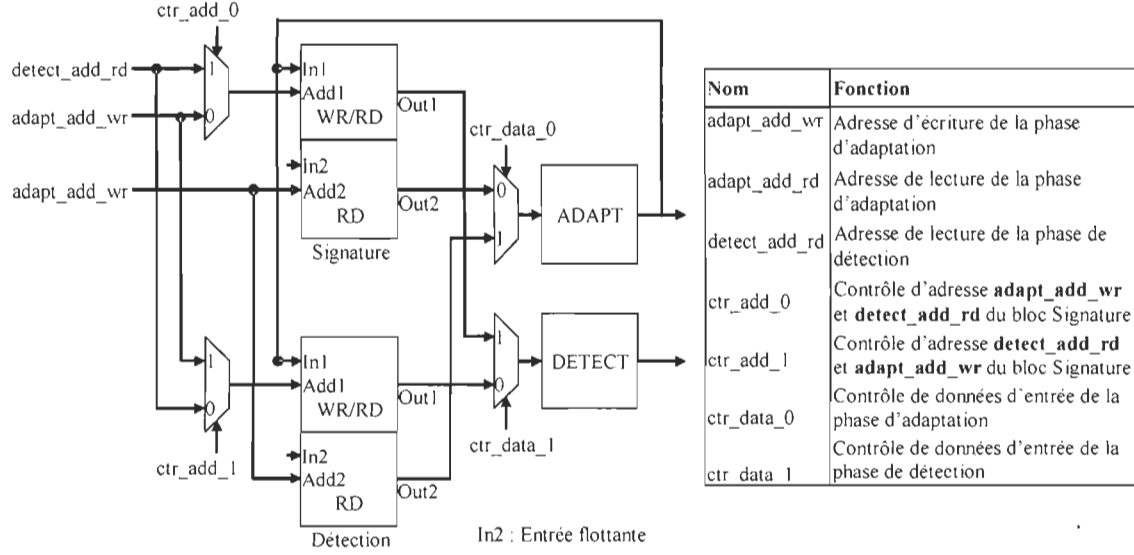
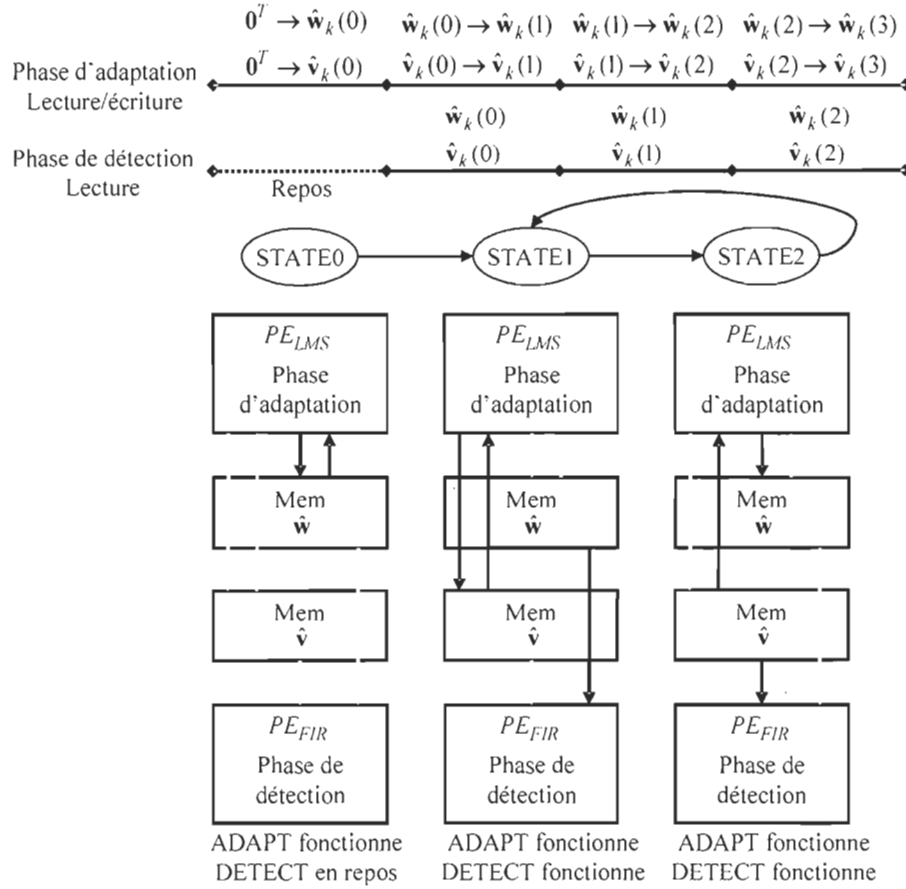


Figure 30 : Structure partielle d'une mémoire semi-globale des poids adaptés ($\hat{\mathbf{w}}$ et $\hat{\mathbf{v}}$). *ADAPT* : phase d'adaptation, *DETECT* : phase de détection, Signature : bloc Signature, Détection : bloc Détection

Un autre type de mémoire consiste en une mémoire tampon *Serial2Parallel* et *Parallel2Serial*. Le tampon convertisseur *Serial2Parallel* sert à temporiser les données en entrée et les convertir en parallèle. Les données en entrée sont respectivement r^{train} pour l'adaptation et \tilde{r} pour la détection. La conversion des données en série fournit les entrées à traiter des usagers aux processeurs élémentaires. Le tampon convertisseur *Parallel2Serial* sert à temporiser les sorties estimées des usagers et les convertir en série. La conversion des données en série a pour but de réduire le nombre des entrées/sorties. Ceci est nécessaire pour l'interface de l'architecture avec l'extérieur. Il est notamment primordial dans le cas où l'architecture est intégrée comme une puce dont les pattes d'entrée/sortie du boîtier sont directement reliés avec les entrées/sorties de l'architecture.



La figure 32 présente le fonctionnement du convertisseur *Parallel2Serial*. L'entrée « input » consiste en une bande en parallèle des entrées, alors que la sortie « output » consiste en séquence de données en série.

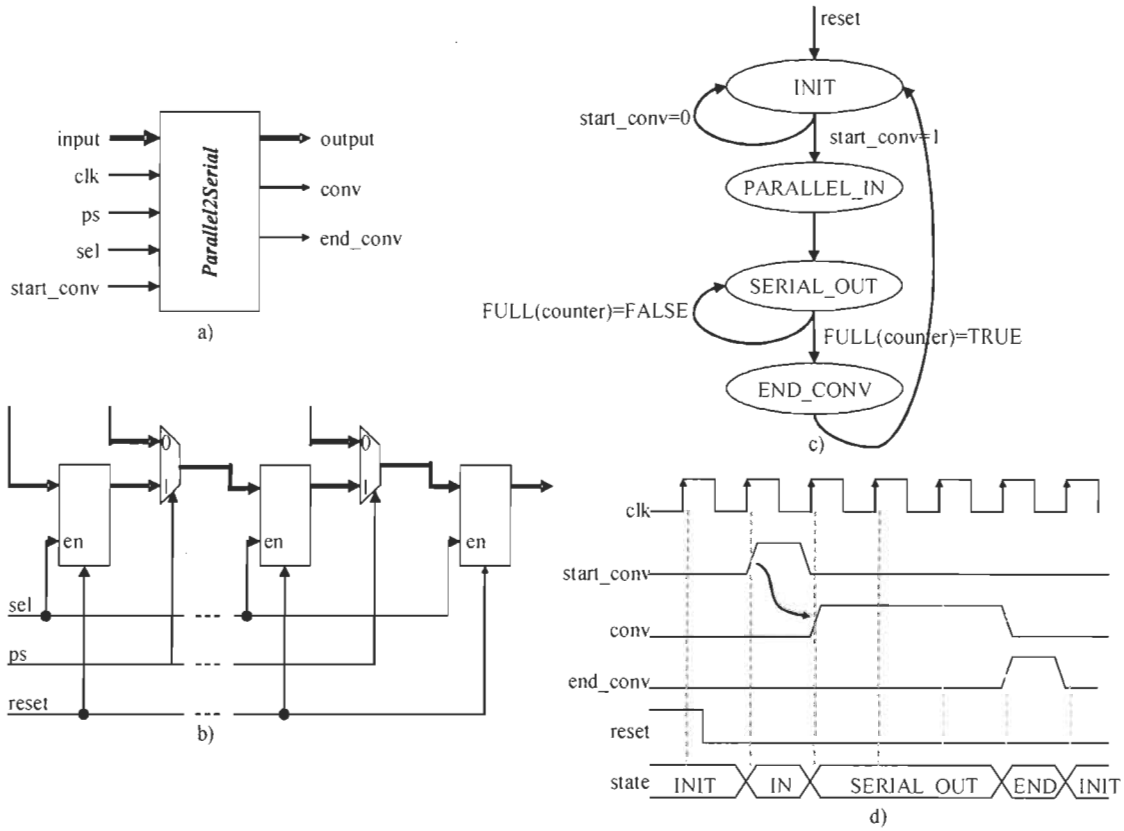


Figure 32 : Structure du convertisseur Parallel2Serial : a) Diagramme bloc, b) Schéma RTL, c) Machine à états, d) Diagramme des états.

III.7. Architecture de l'unité de traitement PE

Les cœurs des *PE* sont les filtres opérant sur des données à valeurs complexes basés sur un multiplieur-accumulateur (MAC) pipeline à deux niveaux. Il s'agit des PE_{LMS} traitant le filtrage adaptatif FIR-LMS et les PE_{LMS} traitant le filtrage transversal FIR.

Considérons un filtre adaptatif FIR-LMS à données complexes exprimé comme suit :

$$\hat{y}(n) = \mathbf{r}(n) \mathbf{w}^H(n) \quad (52)$$

$$e(n) = b^{train}(n) - \hat{y}(n) \quad (53)$$

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \lambda e^*(n) \mathbf{r}(n) \quad (54)$$

où $\tilde{\mathbf{r}}(n)$ dénote le vecteur du signal bruité à reconstruire correspondant au symbole n ;

$\hat{\mathbf{w}}(n)$ dénote le vecteur des poids adaptés du filtre correspondant au symbole n ;

$\hat{y}(n)$ dénote le symbole n reconstruit par le filtre FIR ;

$e^*(n)$ dénote la conjuguée de l'erreur d'adaptation $e(n)$.

L'élément conjugué $e^*(n)$ s'introduit car les données sont complexes [47]. Certes, on peut considérer que l'équation (52) exprime la fonction de filtrage transversal FIR. Aussi, la fonction de filtrage adaptatif FIR-LMS est exprimée par le filtrage transversal (52) et l'adaptation LMS (53-54).

En développant la multiplication à valeurs complexes, le filtre transversal FIR à données complexes peut s'exprimer comme suit :

$$\Re(\hat{y}(n)) = \sum_{i=1}^{N_f} \left(\Re(\tilde{r}^{(n)}(i)) \Re(\hat{w}^{(n)}(i)) - \Im(\tilde{r}^{(n)}(i)) \Im(\hat{w}^{(n)}(i)) \right) \quad (55)$$

$$\Im(\hat{y}(n)) = \sum_{i=1}^{N_f} \left(\Re(\tilde{r}^{(n)}(i)) \Im(\hat{w}^{(n)}(i)) + \Im(\tilde{r}^{(n)}(i)) \Re(\hat{w}^{(n)}(i)) \right) \quad (56)$$

où N_f est la longueur du filtre; \Re et \Im sont respectivement la partie réelle et imaginaire.

L'erreur d'adaptation s'exprime comme suit :

$$\Re(e(n)) = b^{train}(n) - \Re(\hat{y}(n)) \quad (57)$$

$$\Im(e(n)) = b^{train}(n) - \Im(\hat{y}(n)) \quad (58)$$

Dans le cas où on ne s'intéresse qu'à la partie réelle de l'erreur d'adaptation, on ne considère que l'équation (57). Cette simplification est valide dans le cadre de cette thèse étant donné que les données à reconstruire sont de type BPSK. Du coup, le retour d'adaptation de l'algorithme LMS peut s'exprimer comme suit :

$$\Re(\hat{w}^{(n+1)}(i)) = \Re(\hat{w}^{(n)}(i)) + \mu e \Re(\tilde{r}^{(n)}(i)), i = 1, \dots, N_f \quad (59)$$

$$\Im(\hat{w}^{(n+1)}(i)) = \Im(\hat{w}^{(n)}(i)) + \mu e \Im(\tilde{r}^{(n)}(i)), i = 1, \dots, N_f \quad (60)$$

où $e(n) = \Re(e(n))$.

Ainsi, le noyau du processeur PE_{FIR} consiste à réaliser les équations (55-56), alors que le noyau du processeur PE_{LMS} consiste à réaliser les équations (55-56, 57, 59-60).

La figure 33 présente le flot d'états du noyau de filtrage transversal FIR à données complexes. On considère que tous les opérateurs arithmétiques sont à un cycle horloge, sauf que les multiplieurs sont à 2-cycle d'horloge. Le flot est constitué de 11 états $\{Init, S1, \dots, S9, End_FIR\}$. L'état *Init* consiste à initialiser tous les états internes du processeur, incluant ses registres internes. À l'état *End_FIR*, le symbole n reconstruit est disponible pour tout traitement subséquent tel que l'adaptation par LMS, l'écriture aux mémoires, etc. La mise à jour des entrées (lectures des entrées) vise à lire les données bruitées à reconstruire ($\Re(\tilde{r}^{(n)}(i))$, $\Im(\tilde{r}^{(n)}(i))$) et les poids adaptés du filtre ($\Re(\hat{w}^{(n)}(i))$, $\Im(\hat{w}^{(n)}(i))$). Les 4 états (*S6*, *S7*, *S8*, *S9*) ont pour but de finaliser les 2 derniers résultats de la boucle lorsque le compteur atteint la longueur du filtre.

La figure 34 présente le flot d'états du noyau de l'adaptation par LMS à données complexes avec l'erreur d'adaptation réelle. Aussi, on considère que tous les opérateurs arithmétiques sont à un cycle horloge, sauf que les multiplieurs sont à 2 cycles d'horloge. Le flot est constitué de 8 états $\{End_LMS, S1_0, \dots, S6_0, End_Adapt_LMS\}$.

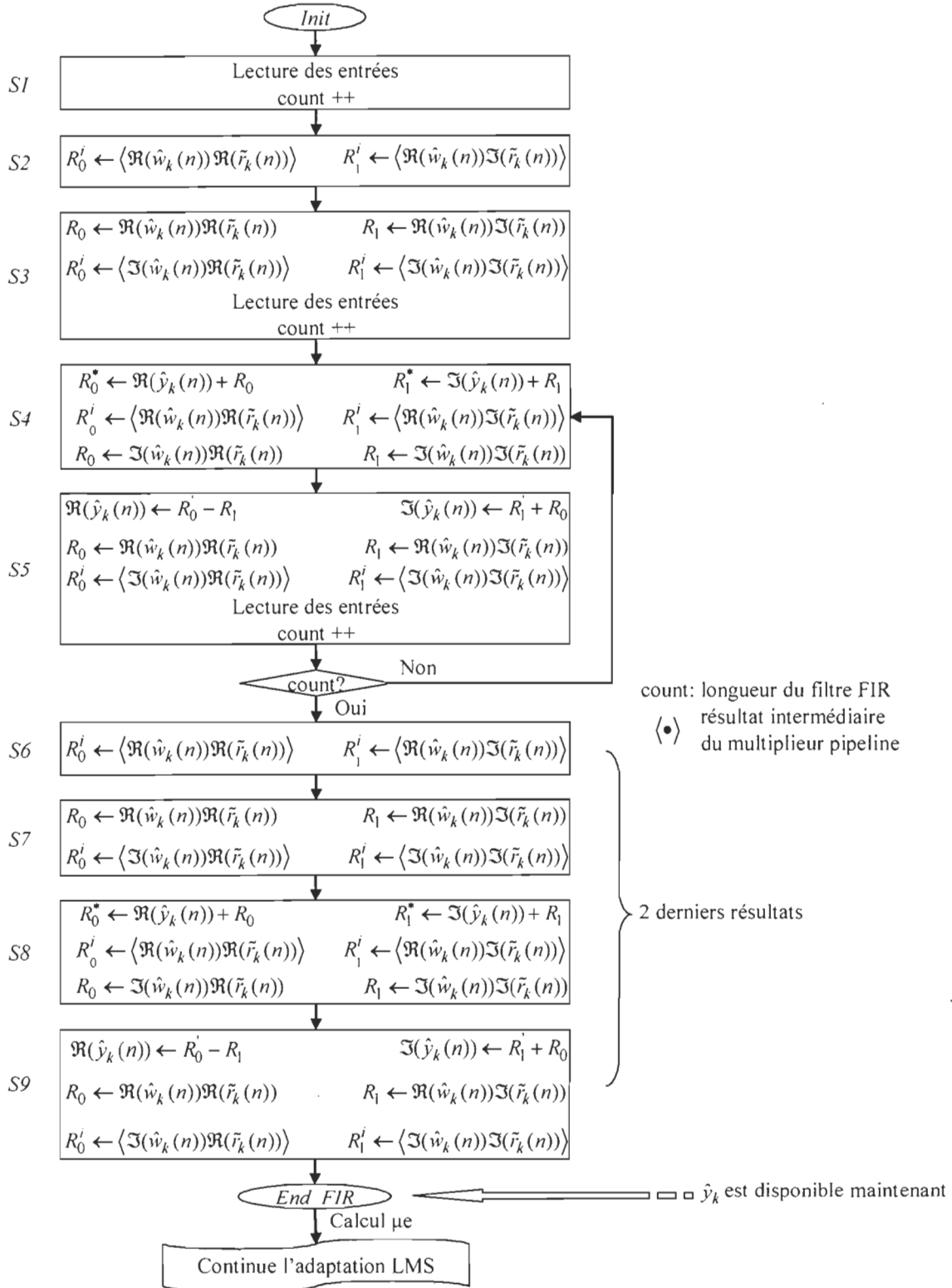


Figure 33 : Flot d'états du noyau FIR à données complexes.

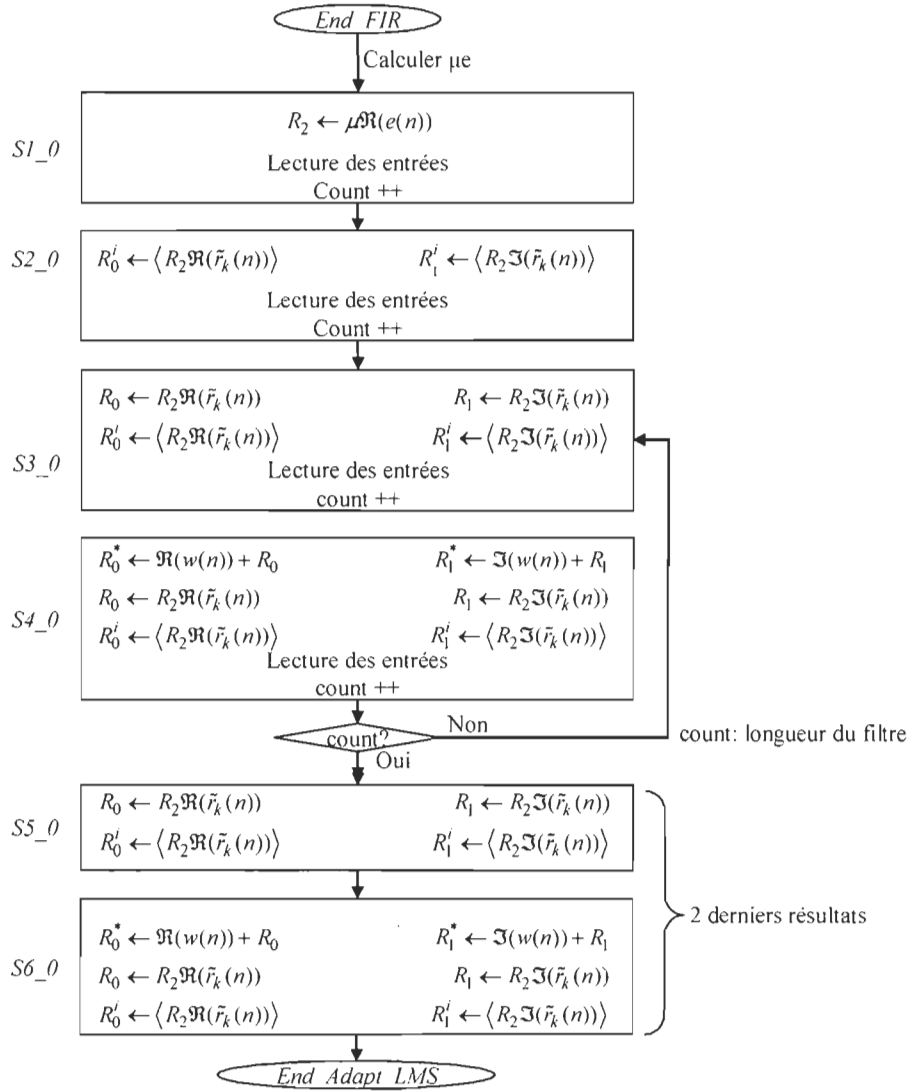


Figure 34 : Flot d'états du noyau de l'adaptation LMS à données complexes avec erreur d'adaptation réelle.

Nous avons conçu un processeur élémentaire capable de traiter à la fois le filtrage transversal et le filtrage adaptatif FIR-LMS. La figure 35 présente le graphe de données des filtres à données complexes. Nous utilisons deux multiplieurs pipelines à deux niveaux et trois additionneurs à un cycle. L'utilisation de ces multiplieurs pipelines nous a permis de concevoir l'architecture pipeline à gras fins du PE au niveau des opérateurs.

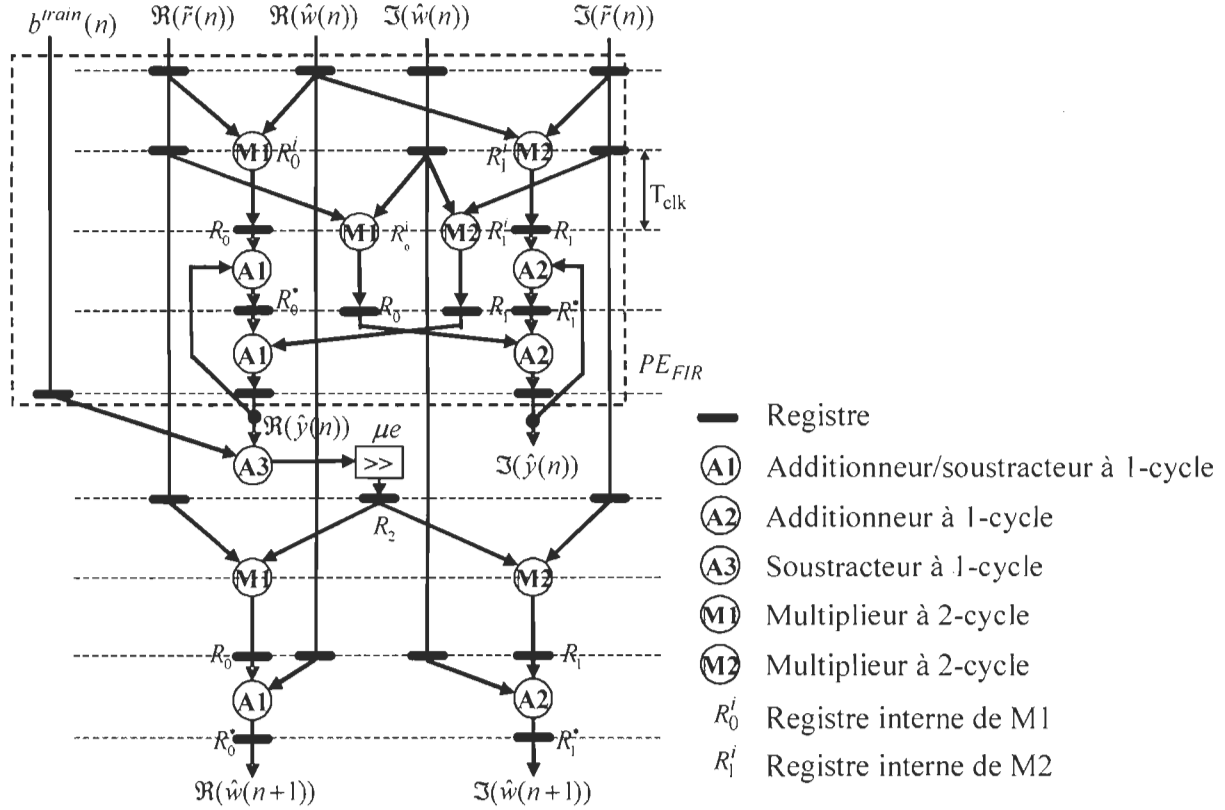


Figure 35 : Graphe du flot de données du processeur LMS à données complexes.

La figure 36 présente le schéma au niveau de transfert de registres (*RTL – Register Transfer Level*). L'architecture se base sur des multiplexeurs. Ces derniers sont contrôlés par un contrôle local.

L'ordonnancement (*scheduling*) consiste à organiser ces PE de façon optimisée. Les contraintes de l'ordonnancement doivent exploiter la structure régulière des FPGA cibles tout en minimisant le nombre d'additionneurs et le schéma critique. Pour ce faire, deux types d'ordonnancement standards [73] sont applicables : ASAP (*As Soon As Possible*) et ALAP (*As Late As Possible*). Tandis que la méthode ASAP offre le délai optimal mais introduit beaucoup de surplus matériel, la méthode ALAP offre le coût matériel optimal mais introduit un certain surplus de latence. Ainsi, on adopte une méthode d'ordonnancement en combinant ces deux méthodes standards.

Le pas d'adaptation μ , avec $0 < \mu < 1$, est choisi pour qu'il soit de 2^n où n est un nombre entier négatif. Le choix de μ s'effectue par essai-erreur expérimentalement en simulation de telle sorte que les performances algorithmiques soient satisfaisantes. La valeur de 2^n permet de réaliser la multiplication de μ par un simple décalage à droite sans avoir besoin d'aucun matériel. Par exemple, une multiplication d'un nombre avec 2^{-5} (0,03125) s'effectue simplement par un décalage de 5-bits à droite.

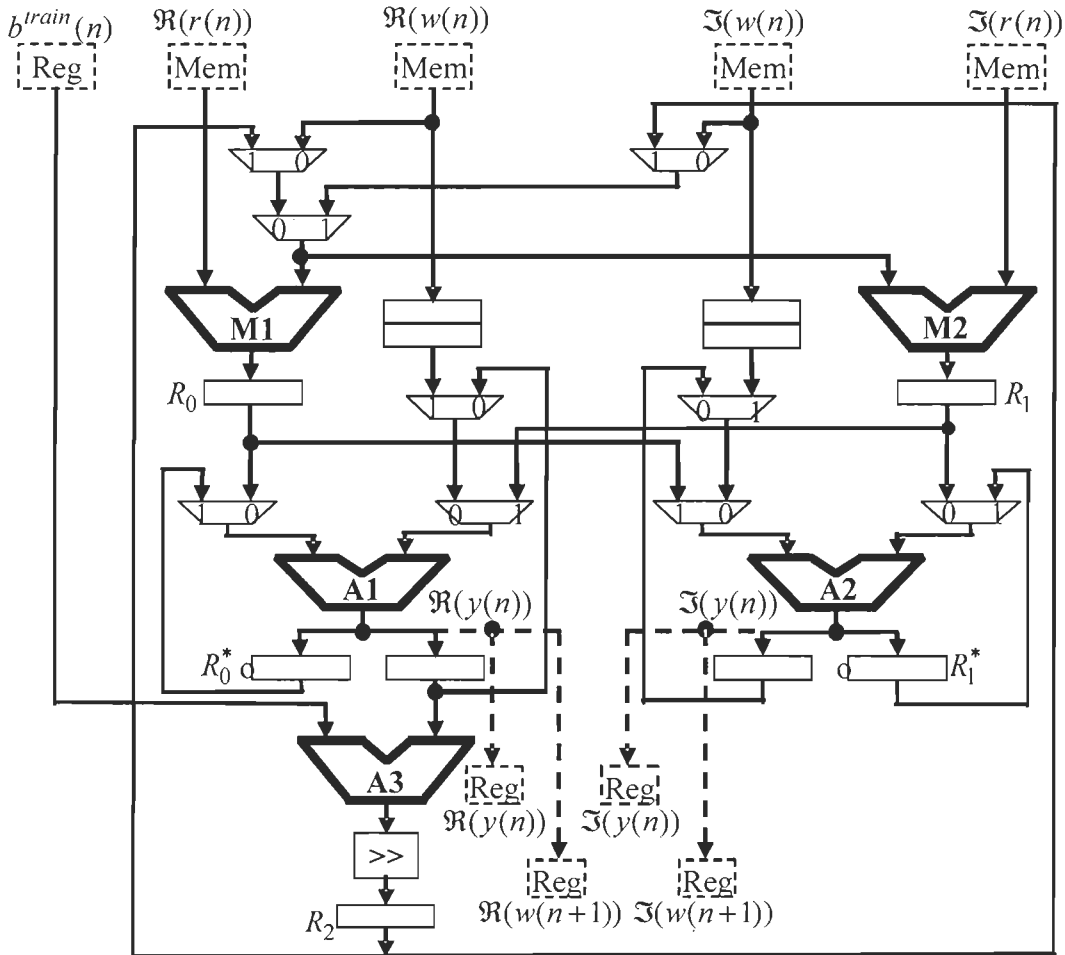


Figure 36 : Schéma RTL du processeur LMS à données complexes.

Le tableau 5 montre le nombre de cycles d'horloge nécessaires pour traiter une fonction du PE. Dans ce tableau, N_f dénote la longueur du filtre à réaliser. Ainsi, la latence d'une fonction de filtrage FIR à données complexes réalisée par ce PE est de 5 cycles. La latence d'une fonction

d'adaptation LMS à données complexes réalisée par ce PE est de 4 cycles. La latence d'une fonction de filtrage adaptatif FIR-LMS à données complexes réalisée par ce PE est de 9 cycles. Le débit d'entrée et sortie de ce PE pour réaliser une de ces trois fonctions – filtrage transversal FIR, adaptation LMS et filtrage adaptatif FIR-LMS à données complexes – sont de deux cycles et d'un cycle respectivement.

Tableau 5 : Nombre de cycles d'horloge nécessaires d'un PE.

Fonction	Nombre de cycles d'horloge nécessaire
Fonction de filtrage transversal FIR	$2N_f + 5$
Fonction d'adaptation LMS	$2N_f + 4$
Fonction de filtrage adaptatif FIR-LMS	$4N_f + 9$

III.8. Parallélisme et stratégie de pipeline

L'architecture consiste en tableaux réguliers de PE qui opèrent en parallèle. Le nombre de ces éléments de traitement dépend des contraintes de communications : nombre d'utilisateurs K , débits définis par OVSF, temps d'adaptation (t_A) et temps de détection (t_D). Il dépend aussi des contraintes matérielles : taille des mémoires BRAM utilisées. Le nombre total des PE est $(N_1 + N_2)$ filtres adaptatifs et $(N_1 + N_2)$ filtres transversaux FIR où N_1 et N_2 sont respectivement le nombre de PE utilisés par le bloc Signature et le bloc Détection.

Pour accélérer le temps d'exécution, nous utilisons deux types de pipeline : pipeline à gros grains au niveau système et pipeline à grains fins au niveau des opérateurs de multiplication. Nous utilisons les multiplieurs pipelines à deux niveaux. Ce sont les multiplieurs dédiés disponibles sur le dé de silicium de FPGA ciblés. La stratégie de pipeline à gros grains au niveau système est présentée aux figures 37 et 38, respectivement pour la phase d'adaptation et la phase de détection. Lorsque le pipeline est plein, l'adaptation du bloc Signature, l'adaptation du bloc

Détection, la détection du bloc Signature et la détection du bloc Détection s'exécutent en parallèle.

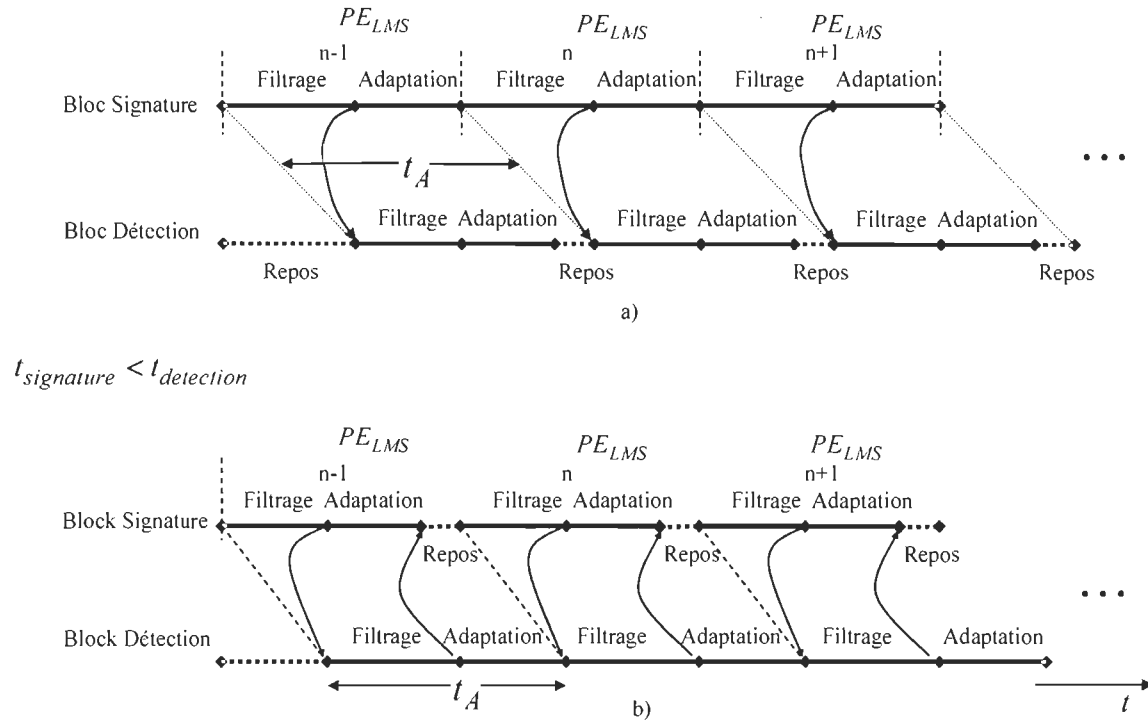


Figure 37 : Stratégie du pipeline de la phase d'adaptation.

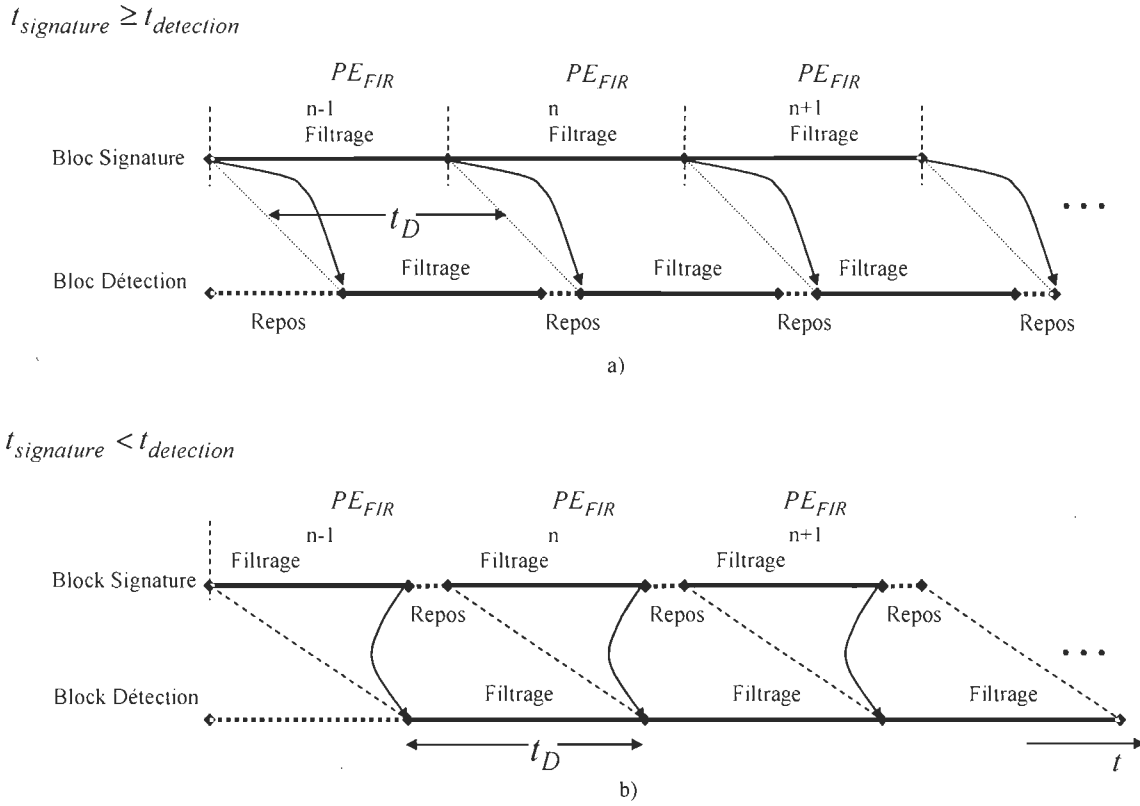


Figure 38 : Stratégie du pipeline de la phase de détection.

D'une manière séquentielle (sans aucun pipeline), le traitement du système suit l'ordre suivant :

L'exécution de la phase d'adaptation du bloc Signature → l'exécution de la phase de détection du bloc Signature → l'exécution de la phase d'adaptation du bloc Détection → l'exécution de la phase de détection du bloc Détection.

Ainsi, le temps d'exécution d'une telle structure séquentielle est le temps total de ces quatre phases.

Avec la stratégie du pipeline des architectures développées, le temps d'adaptation, qui est le temps d'exécution de la phase d'adaptation, est déterminé par la valeur maximale entre temps d'exécution de la phase d'adaptation du bloc Signature et temps d'exécution de la phase d'adaptation du bloc Détection. Également, le temps de détection, qui est le temps d'exécution

de la phase de détection, est déterminé par la valeur maximale entre le temps d'exécution de la phase de détection du bloc Signature et le temps d'exécution de la phase de détection du bloc Détection.

III.9. Résultats d'implémentation

Le but ultime de la mise en œuvre VLSI du récepteur CF-MUD consiste à maximiser le nombre d'utilisateurs qui peut être intégré sur un même composant FPGA. Cet objectif vise donc à améliorer la capacité des BTS en augmentant le nombre d'utilisateurs simultanés actifs tout en minimisant le nombre de composants FPGA. Aussi, augmenter le nombre d'utilisateurs sur un seul composant évite des problèmes éventuels liés à la communication entre composants FPGA.

Les FPGA ciblés sont les composants des familles Virtex-II et Virtex-II Pro de Xilinx®. Les architectures développées sont visées pour des systèmes DS-CDMA asynchrones mono-débit et multi-débit. Les simulations ont été effectuées pour les systèmes DS-CDMA asynchrones. Le canal d'évanouissement considéré est de type Rayleigh à trois coefficients complexes. Le délai relatif maximum entre utilisateurs est de 30 brades, alors que le délai maximum relatif entre trajets d'un même utilisateur est de 10 brades. Dans les systèmes WCDMA, les valeurs de délai maximales se traduisent respectivement en 0,78 ms et 0,26 ms.

Les tableaux 6 et 7 montrent respectivement le nombre maximum d'utilisateurs qui peut être intégré sur un seul composant FPGA de la famille Virtex-II Pro et Virtex-II. L'évanouissement stationnaire, semi-stationnaire et variant se traduit en temps d'adaptation respectif de 40ms, 10 ms et 2ms. Alors que le temps d'adaptation de tous les coefficients de tous les filtres varie selon les conditions d'évanouissement de canal, le temps de détection de tous les symboles de tous les utilisateurs est toujours de 10ms. Dans ces travaux, nous avons considéré 100 itérations pour la phase d'adaptation et $100 \times (256/N_c)$ itérations pour la phase de détection.

Tableau 6 : Nombre maximum d'utilisateurs (\hat{K}) qui peut être intégré sur un seul composant de la famille Virtex-II Pro.

Device OVSF	Stationnaire				Semi-stationnaire				Variant			
	64	16	8	4	64	16	8	4	64	16	8	4
XC2VP2	10	10	8	6	10	6	6	4	4	2	2	1
XC2VP4	22	20	16	14	20	14	10	6	10	4	2	2
XC2VP7	30	28	24	18	28	18	14	8	12	6	4	2
XC2VP20	52	48	36	28	48	28	16	16	22	12	4	2
XC2VP30	68	68	44	32	68	32	26	16	26	12	4	2
XC2VP40	84	82	64	38	82	38	32	16	32	12	4	2
XC2VP50	98	90	68	46	90	46	32	16	38	12	4	2
XC2VP70	112	108	68	64	108	64	32	16	54	12	4	2
XC2VP100	148	136	88	68	136	68	32	16	54	12	4	2
XC2VP125	170	136	110	68	136	68	32	16	54	12	4	2

Tableau 7 : Nombre maximum d'utilisateurs (\hat{K}) qui peut être intégré sur un seul composant de la famille Virtex-II.

Device OVSF	Stationnaire				Semi-stationnaire				Variant			
	64	16	8	4	64	16	8	4	64	16	8	4
XCV80	6	6	6	4	6	4	4	2	2	2	1	1
XCV250	18	18	16	12	18	12	10	6	8	4	2	2
XCV500	24	22	18	16	23	16	10	6	10	4	4	2
XCV1000	28	26	22	16	25	16	12	8	12	6	4	2
XCV1500	34	32	26	20	32	20	16	8	14	8	4	2
XCV2000	36	34	28	22	34	22	16	10	16	9	4	2
XCV3000	56	52	40	32	52	32	19	16	24	12	4	2
XCV4000	66	60	44	32	60	32	24	16	26	12	4	2
XCV6000	72	68	48	32	68	32	28	16	26	12	4	2
XCV8000	84	72	56	32	72	32	32	16	28	12	4	2

Les tableaux 8 et 9 montrent respectivement le taux d'utilisation de ressources matérielles correspondant avec le nombre maximum d'utilisateurs sur les composants des familles Virtex-II Pro et Virtex-II dans ces tableaux, respectivement.

Tableau 8 : Taux d'utilisation de ressources matérielles sur un seul composant de la famille Virtex-II Pro (correspond aux nombres d'utilisateurs reportés au tableau 6).

Device OVSF	Stationnaire				Semi-stationnaire				Variant			
	64	16	8	4	64	16	8	4	64	16	8	4
XC2VP2	93	97	98	88	97	88	100	89	79	83	83	39
XC2VP4	100	100	95	100	100	100	95	100	100	71	57	36
XC2VP7	96	95	98	95	95	95	95	97	98	95	68	23
XC2VP20	98	98	98	99	98	99	97	97	100	82	34	11
XC2VP30	90	100	88	97	100	97	99	94	76	70	22	7
XC2VP40	89	100	100	99	100	99	92	67	100	50	16	5.2
XC2VP50	100	92	98	99	92	99	85	55	98	41	13	4.3
XC2VP70	92	100	83	99	100	99	80	39	99	29	9.1	3.0
XC2VP100	100	92	99	92	92	92	59	29	97	22	6.7	2.2
XC2VP125	92	98	100	98	98	98	47	23	78	17	5.3	1.7

Tableau 9 : Taux d'utilisation de ressources matérielles sur un seul composant de la famille Virtex-II (correspond aux nombres d'utilisateurs reportés au tableau 7).

Device OVSF	Stationnaire				Semi-stationnaire				Variant			
	64	16	8	4	64	16	8	4	64	16	8	4
XCV80	95	98	91	85	98	85	100	88	86	75	58	58
XCV250	98	96	100	90	96	90	97	97	89	83	67	42
XCV500	96	98	99	100	98	100	83	88	87	94	100	31
XCV1000	99	98	92	99	98	99	98	100	90	90	75	25
XCV1500	99	100	98	97	100	97	100	100	97	100	62	21
XCV2000	95	98	100	92	98	92	95	98	95	96	54	18
XCV3000	97	99	100	100	99	100	99	100	100	100	31	10
XCV4000	99	100	100	92	100	92	100	80	87	80	25	8.3
XCV6000	100	94	100	92	94	92	97	89	72	67	21	6.9
XCV8000	100	100	100	79	100	78	98	76	100	57	18	6.0

On suppose que le nombre d'utilisateurs maximum détectable pour chaque débit de données [qui corresponde au facteur OVSF] est égal à son facteur OVSF correspondant. Par exemple, si on considère le facteur OVSF $N_c = 64, 16, 8, 4$, le nombre d'utilisateurs détectables respectif est égale à $K = 64, 16, 8, 4$. Ces facteurs OVSF correspondent respectivement au débit de données de 12.2 kb/s, 64 kb/s, 144 kb/s et 384 kb/s.

Tableau 10 : Nombre de secteurs capable d'être traités sur un seul composant de la famille Virtex-II Pro.

Device OVSF	Stationnaire				Semi-stationnaire				Variant			
	64	16	8	4	64	16	8	4	64	16	8	4
XC2VP2	0	0	0	0	0	0	0	0	0	0	0	0
XC2VP4	1	1	1	0	1	0	0	0	0	0	0	0
XC2VP7	1	1	1	1	1	1	0	0	0	0	0	0
XC2VP20	3	3	2	1	3	1	1	1	1	0	0	0
XC2VP30	3	3	2	2	3	2	1	1	1	0	0	0
XC2VP40	3	3	3	2	3	2	2	1	2	0	0	0
XC2VP50	3	3	3	2	3	2	2	1	2	0	0	0
XC2VP70	3	3	3	3	3	3	2	1	3	0	0	0
XC2VP100	3	3	3	3	3	3	2	1	3	0	0	0
XC2VP125	3	3	3	3	3	3	2	1	3	0	0	0

Tableau 11 : Taux d'utilisation de ressources matérielles pour le nombre de secteurs dans une même BTS du tableau 10 dans les composants de la famille Virtex-II Pro.

Device OVSF	Stationnaire				Semi-stationnaire				Variant			
	64	16	8	4	64	16	8	4	64	16	8	4
XC2VP2	0	0	0	0	0	0	0	0	0	0	0	0
XC2VP4	72	80	95	0	80	0	0	0	0	0	0	0
XC2VP7	51	54	65	84	54	84	0	0	0	0	0	0
XC2VP20	90	98	87	56	98	56	97	97	72	0	0	0
XC2VP30	63	70	64	97	70	97	60	94	46	0	0	0
XC2VP40	50	58	75	83	58	83	92	67	100	0	0	0
XC2VP50	48	49	69	68	49	68	85	55	82	0	0	0
XC2VP70	39	44	58	74	44	74	80	39	88	0	0	0
XC2VP100	32	32	54	64	32	64	59	29	86	0	0	0
XC2VP125	25	34	43	69	34	69	47	23	69	0	0	0

Le fait que le nombre d'utilisateurs maximum possible de ces résultats (dans les tableaux 6 et 7), qui est beaucoup supérieur au nombre d'utilisateurs détectables pour un facteur OVSF donnée, permet d'intégrer plusieurs secteurs d'une même station de base BTS. En effet, il y a trois secteurs, dont chacun couvre 16 usagers, dans une même BTS. Par conséquent, le tableau 10 montre le nombre de secteurs capable d'être traités sur un seul composant de la famille Virtex-II Pro, alors que respectivement le tableau 11 montre le taux d'utilisation de ressources matérielles sur ces composants. Les chiffres zéro dans ces deux tableaux signifient que l'on ne peut pas

traiter la totalité des 16 usagers d'un secteur. Toutefois, un certain nombre d'usagers peut être intégré dans ces cas comme indiqués dans le tableau 6.

Le tableau 12 montre le résultat après le placement & routage à l'aide de l'outil physique de Xilinx (*ISE Foundation*) ciblé sur un composant Virtex-II Pro XCVP30 d'un système dans l'environnement stationnaire et semi-stationnaire.

Tableau 12 : Résultats après placement & routage à l'aide de l'outil physique de Xilinx ciblé sur le composant Virtex-II Pro XC2VP30 pour un système avec $K = 16$ usagers dans l'environnement d'intérieur (stationnaire) et piétons (semi-stationnaire).

Condition d'évanouissement	OVSF	Nb. slice	Nb. BRAM	Nb. multiplieur	Clock Rate (MHz)	Clock skew (ns)	t_A (ms)	t_D (ms)
Stationnaire	64	6149/13696 (44%)	36/136 (32%)	32/136 (23%)	71	0.273	4.53	4.50
	16	4508/13696 (32%)	36/136 (32%)	32/136 (23%)	72	0.271	8.49	13.45
	8	6168/13696 (45%)	56/136 (41%)	52/136 (38%)	74	0.28	4.28	13.10
	4	7474/13696 (54%)	68/136 (50%)	64/136 (47%)	73	0.281	4.192	26.56
Semi-stationnaire	64	6155/13696 (44%)	36/136 (32%)	32/136 (23%)	75	0.279	4.34	4.31
	16	8466/13696 (61%)	68/136 (30%)	64/136 (47%)	83	0.281	3.68	5.82
	8	8493/13696 (61%)	84 (61%)	80 (58%)	49	0.708	8.62	9.89
	4	11940/13696 (87%)	132/136 (97%)	128 (94%)	46	1.181	3.33	20.00

Le système considéré a les caractéristiques ci-dessous.

- Il y a 16 usagers au facteur OVSF $N_c = 16$ (64 kb/s) ;
- Le nombre d'itérations d'adaptation du bloc Signature est de 100 ;
- Le nombre d'itérations d'adaptation du bloc Détection est de $100 \times (256 / N_c) = 400$;

- Le pas d'adaptation (μ) du bloc Signature et du bloc est de 0,03125 ($=2^{-5}$).

Dans le tableau 12, t_A et t_D dénotent respectivement le temps d'adaptation et le temps de détection. Comme défini auparavant, le temps d'adaptation est le temps permis pour adapter tous les poids adaptatifs de tous les filtres du récepteur. Respectivement, le temps de détection est le temps permis pour détecter tous les symboles d'une trame de données de 38400 brides. Tout dépendant de la longueur du facteur OVSF considéré, le nombre de symboles change d'un facteur OVSF à l'autre. En effet, le nombre de symboles (N_b) dans une trame est déterminé comme $N_b = 38400 / (256 / N_c)$, où N_c dénote le facteur OVSF.

Finalement, le tableau 13 présente une comparaison des travaux d'implémentation proposés du récepteur CF-MUD par rapport aux travaux les plus significatifs du tableau 3.

Les travaux proposés sont marqués par la couleur foncée. Il n'est pas facile d'établir une base commune afin de comparer nos travaux par rapport à d'autres propositions. Toutefois, nous avons établi la base concernant le type de récepteur, la technologie cible, la performance algorithmique, la norme visée, la complexité matérielle et la capacité de maximiser le nombre d'utilisateurs sur une seule puce (ou un seul composant FPGA). Ainsi, on peut remarquer qu'une fois l'algorithme devient sophistiqué, la technologie DSP⁹ ne répond plus en terme de complexité de calcul. Le choix de FPGA ou ASIC devient évident. D'autres travaux visent principalement à la validation des algorithmes en matériel, mais pas (ou peu dans les travaux [90]) aux normes de 3G, notamment les contraintes temporelles d'exécution. En plus, la capacité de maximiser le nombre d'utilisateurs capable d'être sur un seul composant s'avère grandement avantageuse en pratique, notamment l'intégration de la fonction MUD dans les systèmes de communication sans fil.

⁹ Nous considérons un seul DSP pour cette étude.

Tableau 13 : Comparaison des travaux proposés par rapport aux travaux d'implémentation du tableau 3.

No.	Travaux	Année	Récepteur MUD	Technologie cible	Performance algorithmique	Norme	Complexité matérielle	Maximiser le nombre d'utilisateurs sur un dé silicium
1	Ching-Hung et al. [26]	2001	MF	ASIC	Très faible	Non-définie	Faible	Non
2	Correal et al. [28]	1999	PIC à 2 étages	DSP	Élevé	Non-définie	Non-définie	Non
3	Eltawil et al. [36]	2004	<i>Rake</i> à 2 <i>fingers</i>	ASIC	Faible	Non-définie	Faible	Non
4	Ho et al. [49]	2006	CF-MUD	FPGA	Élevé	WCDMA	Non-défini	Non
5	TI [54]	2000	<i>Rake</i> à 6 <i>fingers</i>	DSP	Faible	WCDMA	Non-définie	Non
6	Lequepeys et al. [62]	1998	MPIC à 2 étages	ASIC et prototypé sur FPGA	Élevé	Non-définie	Non-définie	Non
7	Leung et al. [63]	2000	<i>Rake</i> à 12 <i>fingers</i>	FPGA	Faible	IS-95	Moyenne	Non
8	Xu et al. [114]	2002	MPIC à 4 étages	ASIC	Élevé	WCDMA	Moyenne	Non
9	Rajagopal et al. [90]	2002	MPIC à 3 étages	ASIC	Élevé	Non-définie	Très élevé	Non

III.10. Conclusions

L'architecture matérielle du CF-MUD a été conçue tout en exploitant la particularité des FPGA cibles. Toutefois, elle est applicable pour d'autres cibles technologiques, voire ASIC, grâce à sa

régularité. L'architecture nous a permis de publier 3 articles, incluant un papier journal [J1], et deux papiers conférences [C3]-[C4]. Le développement matériel de l'architecture consiste en une modélisation VHDL d'environ 6000 lignes de code. Bien que certains composants de l'architecture puissent être conçus et modélisés de façon plus comportementale, nous nous contentons de les faire de façon mixte (comportementale structurelle). L'implémentation matérielle a été validée étape par étape jusqu'à la simulation après placement & routage selon le flot de conception conventionnel (cf. V.2).

Bien que l'objectif de ce travail de thèse consiste principalement en les développements architecturaux, plusieurs travaux de simulations algorithmiques ont été effectués afin de comprendre l'algorithme. Par conséquent, les modèles MATLAB®, C/C++ du récepteur CF-MUD ont été développés. De plus, nous avons modélisé en MATLAB® une plateforme simple afin de simuler les récepteurs à usagers multiples d'intérêt dans l'environnement MATLAB®. Le développement des bibliothèques des éléments de base VHDL nous a motivé à développer un paquetage VHDL à virgule fixe. Également, des simulations à virgule fixe nous ont motivé à développer une bibliothèque en virgule fixe en environnement MATLAB®. Ces bibliothèques font le sujet des travaux dans la section V.4.

CHAPITRE IV

ARCHITECTURES PROPOSÉES DU RÉCEPTEUR MUD MULTI-DÉBIT

Les algorithmes CF-MUD mono-débit offrent des meilleures performances aussi bien au niveau algorithmique qu'au niveau matériel vis-à-vis des récepteurs standards [30]. Les architectures VLSI de ce récepteur mono-débit ont été développées dans le chapitre III. L'objectif de ce présent chapitre consiste à utiliser ces noyaux matériels mono-débit (cf. chapitre III) afin d'implémenter le récepteur MUD pour les systèmes multi-débit. Pour atteindre cet objectif, il est évidemment nécessaire d'étendre l'algorithme CF-MUD mono-débit aux systèmes multi-débit. Ainsi, l'extension et l'implémentation VLSI du récepteur MUD pour les systèmes multi-débit font l'objet de ce chapitre.

IV.1. Extension de l'algorithme CF-MUD aux systèmes multi-débit

Ces travaux d'extension visent à étendre l'algorithme CF-MUD pour les systèmes multi-débit. Le modèle d'un système multi-débit a été exposé au chapitre III et présenté à la figure 15. Pour la simplification des travaux tout en gardant le même apport scientifique, on considère le bloc Signature seul de l'algorithme CF-MUD mono-débit. En plus, comme nous allons le démontrer plus loin (cf. IV.3), le bloc Signature seul offre des performances satisfaisantes [en taux d'erreur binaire BER] par rapport au récepteur *Rake* et SoftMPIC. La notion des systèmes multi-débit utilisée dans ce chapitre a été auparavant définie dans le chapitre II (cf. II.3.1).

Nous considérons les deux schémas multi-débit LRD et HRD. Les schémas multi-débit respectifs sont CF-MUD multi-débit à débit faible (*LRD-CF-MUD : Low Rate Detector CF-MUD*) et CF-MUD multi-débit à débit élevé (*HRD-CF-MUD : High Rate Detector CF-MUD*). Dans le cas où nous considérons la Signature adaptative seule, les schémas multi-débit sont LRD-Sign-MUD et HRD-Sign-MUD respectivement.

Dans les travaux existants, les auteurs ont proposés des codes d'étalement virtuels et des usagers virtuels respectifs à ces derniers [52], [55]-[56], [77]-[78], [93]-[94], [115], [119]. Cette approche, qui construit des codes virtuels, est valable car ces propositions existantes sont basées sur l'arithmétique des matrices des codes : MMSE [77], [93], et décorrélateur [23], [55], [94]. En effet, ces derniers se basent sur le calcul des matrices d'inter-corrélation et d'auto-corrélation des codes signatures [109].

Contrairement à ces travaux existants, nous considérons des codes signatures réels¹⁰ pour construire des usagers virtuels. En effet, l'algorithme CF-MUD n'est pas basé sur le calcul des matrices d'inter-corrélation et d'auto-corrélation des codes signatures. Il se base sur une technique d'adaptation des signatures utilisant les séquences d'apprentissage. Ces séquences d'apprentissage sont à générer à l'intérieur du récepteur puisque nous n'avons pas accès au préalable aux données d'apprentissages des usagers. Nous construisons ainsi les éléments virtuels du signal d'apprentissage.

En effet, comme présenté à la figure 22 au chapitre III, deux entrées du récepteur MUD sont le signal d'apprentissage (\mathbf{r}^{train}) et le signal bruité à estimer ($\tilde{\mathbf{r}}$). Ces deux entrées sont communes à tous les usagers¹¹. Le signal bruité $\tilde{\mathbf{r}}$ est reçu au récepteur à partir des signaux transmis de tous les usagers via le canal de transmission. Nous n'avons pas ainsi de contrôle sur des signaux transmis au niveau du récepteur. Or, il est possible de contrôler le signal d'apprentissage \mathbf{r}^{train} , lequel est intérieurement généré à partir des paramètres estimés des usagers : profil puissance-délai des trajets du canal de transmission et les codes signatures des usagers.

¹⁰ Nous ne dérivons pas les codes signatures virtuels.

¹¹ Mais avec les délais différents.

Ainsi, pour dériver un système mono-débit équivalent, on dérive les usagers virtuels en dérivant les éléments virtuels constituants suivants :

- Les séquences d'entraînement virtuelles des usagers virtuels.
- Les délais relatifs virtuels (entre usagers et entre trajets d'un même usager) à partir des délais réels des usagers réels respectifs.

IV.1.1. Méthode LRD-Sign-MUD

Cette méthode vise à étendre la Signature adaptative pour des systèmes multi-débit en dérivant un système équivalent mono-débit de débit le plus faible. La figure 39 présente le principe du récepteur LRD-Sign-MUD basée sur la Signature adaptative mono-débit qui est le premier bloc constituant du CF-MUD. Dans une période d'observation du système mono-débit équivalent qui est la période des symboles à débit le plus faible (T^l), il y a un symbole de chaque usager à débit de données le plus faible du groupe 1 et N_c^l / N_c^g symboles de chaque usager à débit élevé du groupe g .

Dans cette figure 39, on retrouve quatre blocs constituants du récepteur LRD-Sign-MUD :

- Génération des séquences d'entraînement.
- Dérivation des usagers virtuels.
- Bloc des $K^l + \sum_{g \neq 1} \frac{N_c^l}{N_c^g} K^g$ Signatures adaptatives mono-débit.
- Réorganisation des données.

Le bloc « Génération des séquences d'entraînement » consiste à générer les séquences BPSK b^{train} . Ces dernières sont utilisées comme les séquences désirées pour l'adaptation des filtres des signatures adaptatives. Elles sont aussi utilisées pour dériver les usagers virtuels. Les

usagers virtuels se caractérisent par les délais virtuels des trajets du canal de transmission et les séquences d'entraînement virtuelles des usagers virtuels.

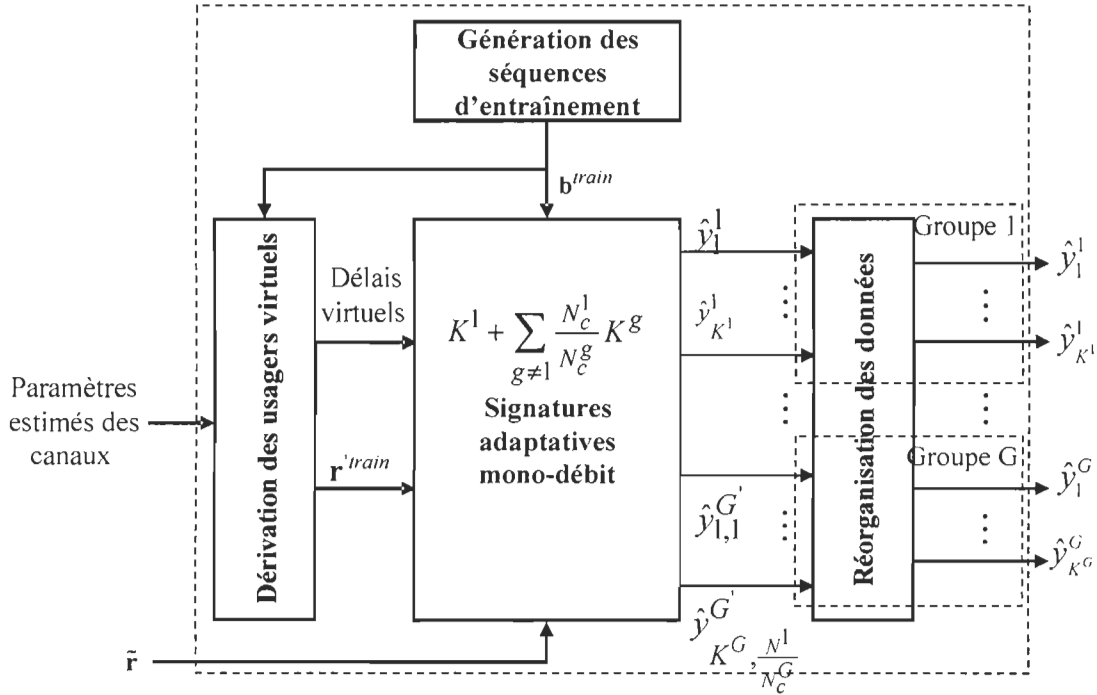


Figure 39: Principe du LRD-Sign-MUD basé sur la Signature adaptative mono-débit. Le récepteur fonctionne au niveau de brides avec période des symboles N^1 .

Les séquences d'entraînement virtuelles s'expriment comme suit :

$$r_{k,v}^{g^{train}(n)}(t) = \begin{cases} r_k^{g^{train}(n)}(t), vT_c^g \leq t \leq (v+1)T_c^g, \text{ où } 1 \leq v \leq \frac{N_c^1}{N_c^g} \\ 0, \text{ ailleurs} \end{cases} \quad (61)$$

où $r_{k,v}^{g^{train}(n)}(t)$ est la séquence des données d'entraînement de l'utilisateur virtuel dérivée de la séquence de données d'entraînement de l'utilisateur réel k du groupe g ($r_k^{g^{train}(n)}(t)$).

Ce principe de génération des séquences d'entraînement virtuelles est semblable à celui présenté à la figure 17. Ces séquences $r_{k,v}^{g^{train}(n)}(t)$ sont utilisées, avec les délais virtuels

respectifs, pour générer les données d'entrée pour l'adaptation des signatures adaptatives (\mathbf{r}^{train}).

Le signal \mathbf{r}^{train} s'exprime alors comme suit :

$$\mathbf{r}^{train}(t) = \sum_{g=1}^G \left[\sum_{k=1}^{K^g} \left(\sum_{v=1}^{\frac{N_c^l}{N_c^g}} r_{k,v}^{g,train}(t) \right) \right] \quad (62)$$

Les délais virtuels relatifs entre usagers et les délais virtuels relatifs entre trajets d'un même usager k du group g sont respectivement définis comme suit :

$$\tau_{k,v}^{g'} = \tau_k^g + vT^g, v = 1, \dots, \frac{N_c^l}{N_c^g} \quad (63)$$

$$\tau_{k,v,l}^{g'} = \tau_{k,l}^g + vT^g, v = 1, \dots, \frac{N_c^l}{N_c^g} \quad (64)$$

où $\tau_{k,v}^{g'}$ dénote le délai relatif virtuel dérivé de celui de l'utilisateur k du groupe g (τ_k^g) ;

$\tau_{k,v,l}^{g'}$ dénote le délai relatif virtuel dérivé de celui du trajet l de l'utilisateur k du groupe g ($\tau_{k,l}^g$).

Le bloc « Signatures adaptatives mono-débit » est le même que le bloc Signature décrit dans la section III.1.1 du chapitre III. Il consiste ainsi en $K^1 + \sum_{g \neq 1} \frac{N_c^l}{N_c^g} K^g$ Signatures dont chacune a pour but d'estimer un usager virtuel dérivé. La longueur des filtres est $N_f = 2N^1$ où N_c^1 dénote le facteur OVFS du groupe de débit le plus faible.

Les sorties estimées de ce bloc s'expriment comme suit :

$$\hat{\mathbf{y}} = \left[\hat{\mathbf{y}}^{T^1(1)}, \hat{\mathbf{y}}^{T^1(2)}, \dots, \hat{\mathbf{y}}^{T^1(N_b)} \right] \quad (65)$$

$$\text{où } \hat{\mathbf{y}}^{T^1(n)} = \left[\hat{y}_1^{l(n)}, \dots, \hat{y}_{K^1}^{l(n)}, \dots, \hat{y}_{1,1}^{g(n)}, \dots, \hat{y}_{1, \frac{N_c^1}{N_c^g}}^{g(n)}, \dots, \hat{y}_{K^g, 1}^{g(n)}, \dots, \hat{y}_{K^g, \frac{N_c^1}{N_c^g}}^{g(n)}, \dots, \hat{y}_{K^G, 1}^{G(n)}, \dots, \hat{y}_{K^G, \frac{N_c^G}{N_c^g}}^{G(n)} \right];$$

N_b dénote le nombre de symboles à estimer.

Le bloc « Réorganisation des données » vise à réorganiser les sorties des usagers à débits élevés afin que leurs index temporelles des données estimées soient correctes. Étant donné le seul débit dans le système équivalent, à chaque fréquence d'échantillonnage des symboles correspondant au facteur OVSF N_c^1 à la sortie, il y a un symbole estimé de chaque usager de débit le plus faible N_c^1 / N_c^g et symboles estimés de chaque usager du groupe g .

Finalement, les symboles estimés de chaque usager de chaque groupe de débit sont obtenus comme suit :

$$\hat{b}_k^g(n) = \text{sign}(\hat{y}_k^g(n)) \quad (66)$$

où $\hat{b}_k^g(n)$ dénote le symbole n estimé de l'utilisateur k du groupe g ;

$\hat{y}_k^g(n)$ dénote sa valeur douce estimée par le récepteur ; et

la fonction $\text{sign}(\bullet)$ consiste simplement en une décision dure.

IV.1.2. Méthode HRD-Sign-MUD

Cette méthode vise à étendre la Signature adaptative seule pour des systèmes multi-débit en dérivant un système équivalent mono-débit de débit le plus élevé. La figure 40 présente le principe du récepteur HRD-Sign-MUD basée sur la Signature adaptative mono-débit qui est le premier bloc constituant du CF-MUD. Dans une période d'observation du système mono-débit équivalent qui est la période des symboles à débit le plus élevé (T^G), il y a un symbole de

chaque usager à débit de données le plus élevé du group T^G et une partie d'un symbole (N_c^g / N_c^G) de chaque usager à débit élevé du groupe g .

Dans cette figure 40, on retrouve quatre blocs constituant du récepteur HRD-Sign-MUD :

- Génération des séquences d'entraînement.
- Dérivation des usagers virtuels.
- Bloc des $\sum_{g=1}^G K^g$ Signatures adaptatives mono-débit.
- Combinaison MRC.

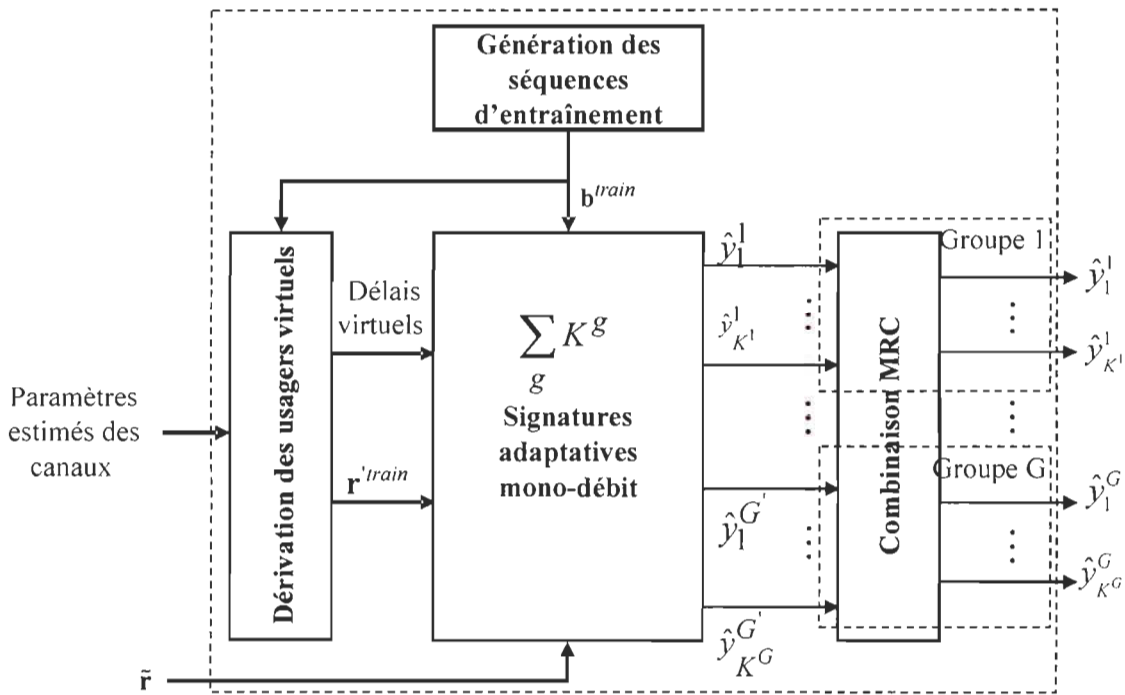


Figure 40 : Principe du HRD-Sign-MUD basé sur la Signature adaptative mono-débit. Le récepteur fonctionne au niveau de brides avec période des symboles N_c^G .

Le bloc « Génération des séquences d'entraînement » consiste à générer les séquences BPSK b^{train} . Ces dernières sont utilisées comme les séquences désirées pour l'adaptation des filtres des signatures adaptatives. Elles sont aussi utilisées pour dériver les usagers virtuels. Les usagers virtuels se caractérisent par les délais virtuels des trajets du canal de transmission et les séquences d'entraînement virtuelles des usagers virtuels.

Les séquences d'entraînement virtuelles s'expriment comme suit :

$$r_k^{g'train(n)}(t) = r_k^{gtrain(n)}(t), nT_c^G \leq t \leq (n+1)T_c^G \quad (67)$$

où $r_k^{g'train(n)}(t)$ est la séquence des données d'entraînement de l'utilisateur virtuel dérivée de la séquence de données d'entraînement de l'utilisateur réel k du groupe g ($r_k^{gtrain(n)}(t)$).

Ce principe qui consiste à générer les séquences d'entraînement virtuelles est semblable à celui de la figure 17. Ces séquences $r_k^{g'train(n)}(t)$ sont utilisées, avec les délais virtuels respectifs, pour générer l'entrée pour l'adaptation des signatures adaptatives (\mathbf{r}'^{train}).

Le signal \mathbf{r}'^{train} s'exprime alors comme suit :

$$\mathbf{r}'^{train}(t) = \sum_{g=1}^G \left(\sum_{k=1}^{K^g} r_k^{g'train}(t) \right) \quad (68)$$

Les délais virtuels relatifs entre usager et les délais virtuels relatifs entre trajets d'un même usager k du groupe g sont respectivement définis comme suit :

$$\tau_k^{g'} = \tau_k^g \quad (69)$$

$$\tau_{k,l}^{g'} = \tau_{k,l}^g \quad (70)$$

où $\tau_k^{g'}$ dénote le délai relatif virtuel dérivé de celui de l'utilisateur k du groupe g (τ_k^g) ; et

$\tau_{k,l}^{g'}$ dénote le délai relatif virtuel dérivé de celui du trajet l de l'utilisateur k du groupe l ($\tau_{k,l}^g$).

Ainsi, on peut constater que ces valeurs virtuelles ($\tau_k^{g'}, \tau_{k,l}^{g'}$) ne changent pas par rapport à celles des usagers réels originaux (69-70).

Le bloc « Signatures adaptatives mono-débit » est le même que le bloc Signature décrit dans la section III.1.1 au chapitre III. Il consiste ainsi en $\sum_{g=1}^{g=G} K^g$ Signatures dont chacune a pour but d'estimer un usager virtuel dérivé. La longueur des filtres est $N_f = 2N_c^G$ où N_c^G dénote le facteur OVFSF du groupe de débit le plus élevé.

Les sorties estimées de ce bloc s'expriment comme suit :

$$\hat{\mathbf{y}} = [\hat{\mathbf{y}}^{T^G(1)}, \hat{\mathbf{y}}^{T^G(2)}, \dots, \hat{\mathbf{y}}^{T^G(N_b)}] \quad (71)$$

où $\hat{\mathbf{y}}^{T^G(n)} = [\hat{y}_1^{l(n)}, \dots, \hat{y}_{K^l}^{l(n)}, \dots, \hat{y}_1^{g'(n)}, \dots, \hat{y}_{K^{g'}}^{g'(n)}, \dots, \hat{y}_1^{G(n)}, \dots, \hat{y}_{K^G}^{G(n)}]^T$;

N_b dénote le nombre de symboles à estimer.

Le bloc « Combinaison MRC » vise à combiner les sorties douces des usagers à débits faibles. Étant donné le seul débit dans le système équivalent, à chaque fréquence d'échantillonnage des symboles correspondant au facteur OVFSF N_c^G à la sortie, il y a un symbole estimé de chaque usager de débit le plus élevé et une partie d'un symbole (N_c^g / N_c^G) de chaque usager du groupe g .

La combinaison MRC s'effectue ainsi comme suit :

$$\hat{y}_k^g(n) = \sum_{v=1}^{\frac{N_c^g}{N_c^G}} \hat{y}_{k,v}^{g'}(n+v). \quad (72)$$

Finalement, les symboles estimés de chaque usager de chaque groupe de débit sont obtenus comme suit :

$$\hat{b}_k^g(n) = \text{sign}(\hat{y}_k^g(n)) \quad (73)$$

où $\hat{b}_k^g(n)$ dénote le symbole n estimé de l'utilisateur k du groupe g ;

$\hat{y}_k^g(n)$ dénote la valeur douce estimée par le récepteur ; et

la fonction $\text{sign}(\bullet)$ consiste simplement en une décision dure.

Dans la suite, nous décrivons les deux méthodes multi-débit que nous avons développé en se basant sur le noyau de la Signature adaptative.

IV.2. Architecture matérielle des récepteurs multi-débit

IV.2.1. Architecture du récepteur LRD-Sign-MUD

Le noyau matériel de la Signature adaptative est utilisé pour développer l'architecture VLSI du récepteur LRD-Sign-MUD. Ainsi, la figure 41 présente l'architecture du LRD-Sign-MUD.

Dans la phase d'adaptation et la phase de détection, le processeur élémentaire PE déjà conçu au chapitre III (cf. III.7) est utilisé pour implémenter les Signatures des usagers virtuels. En plus, l'architecture de Signature adaptative mono-débit est la même que celle du bloc Signature développé dans le chapitre III.

Les délais relatifs des usagers virtuels sont utilisés pour initialiser l'adresse des mémoires. La fonction $\text{sign}(\bullet)$ est implémentée avant la réorganisation des sorties pour réduire davantage la longueur des sorties à l'extérieur. En effet, il suffit d'un bit pour coder chaque sortie après cette fonction de décision dure.

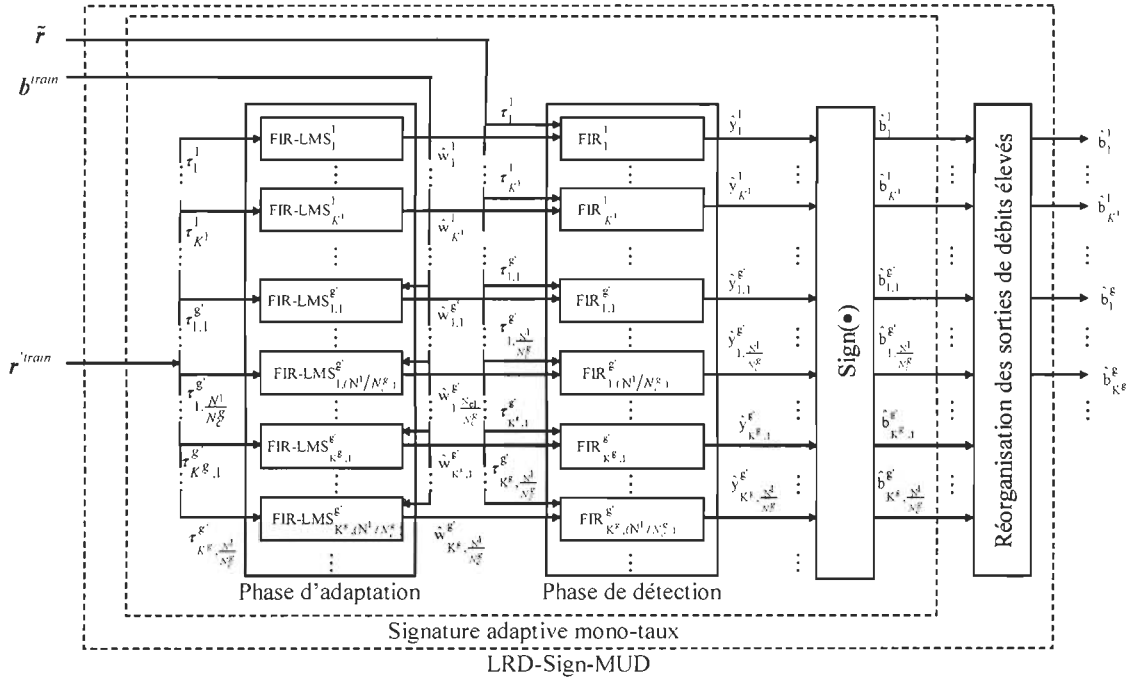


Figure 41 : Architecture du récepteur LRD-Sign-MUD basée sur la Signature adaptative mono-débit.

IV.2.2. Architecture du récepteur HRD-Sign-MUD

Le noyau matériel de la Signature adaptative est utilisé pour développer l'architecture VLSI du récepteur HRD-Sign-MUD. Ainsi, la figure 42 présente l'architecture du HRD-Sign-MUD.

Comme le récepteur LRD-Sign-MUD dans la section IV.1.1, dans la phase d'adaptation et la phase de détection, le processeur élémentaire PE déjà conçu au chapitre III (cf. III.7) est utilisé pour implémenter les signatures des usagers virtuels. En plus, l'architecture de Signature adaptative mono-débit est la même que celle du bloc Signature développé dans le chapitre III. Les délais relatifs des usagers virtuels sont utilisés pour initialiser l'adresse des mémoires.

Aussi, la fonction $sign(\bullet)$ est implémentée après la combinaison MRC pour réduire davantage la longueur des sorties à l'extérieur. En effet, il suffit d'un bit pour coder chaque sortie après cette fonction de décision dure.

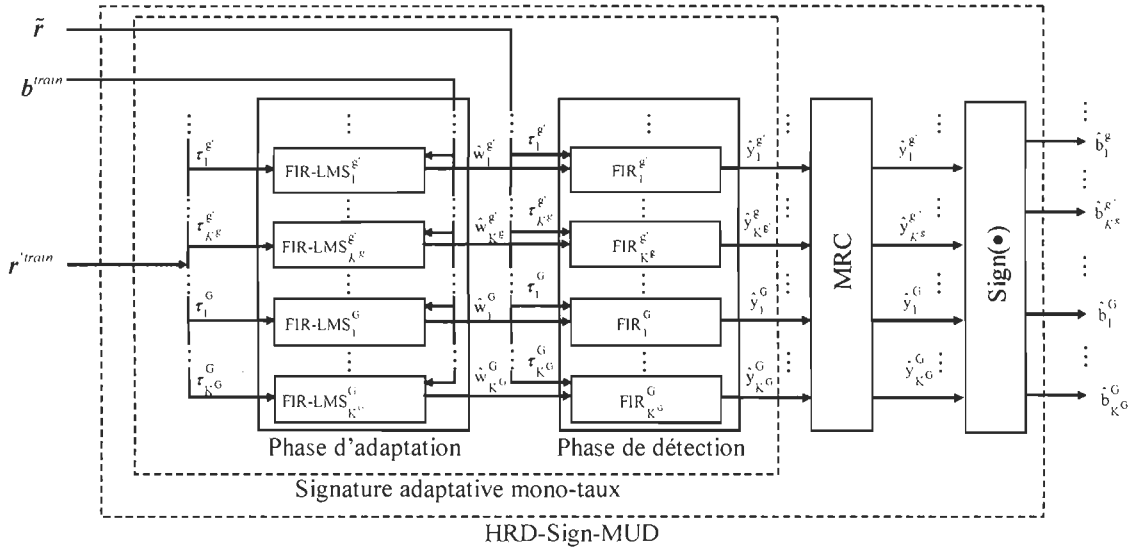


Figure 42 : Architecture du récepteur HRD-Sign-MUD basée sur la Signature adaptative mono-débit.

La combinaison MRC peut être implémentée par une des quatre structures présentées à la figure 43. Dans ces structures, la MRC est implémentée par les additionneurs binaires à deux entrées. Dans cette figure, les structures en a) et c) sont non-pipelinnées alors que les structures en b) et d) sont pipelinnées. Supposons le même délai de propagation¹² pour tous les additionneurs – dénoté par Δ , la latence de ces trois structures est respectivement comme suit :

- $\Delta \log_2 N_b$ pour la structure en a),
- Δ pour la structure en b) et d), et
- $\Delta(N_b - 1)$ pour la structure en c).

où N_b dénote le nombre d'entrées.

La structure en c) est comme la structure RCA (*Ripple Carry Adder*) de l'additionneur.

¹² En réalité, le délai de propagation de l'additionneur n'est pas constant. Il peut changer en fonction de l'activité de circuit qui est déterminée par les données d'entrées et le type de codage de données.

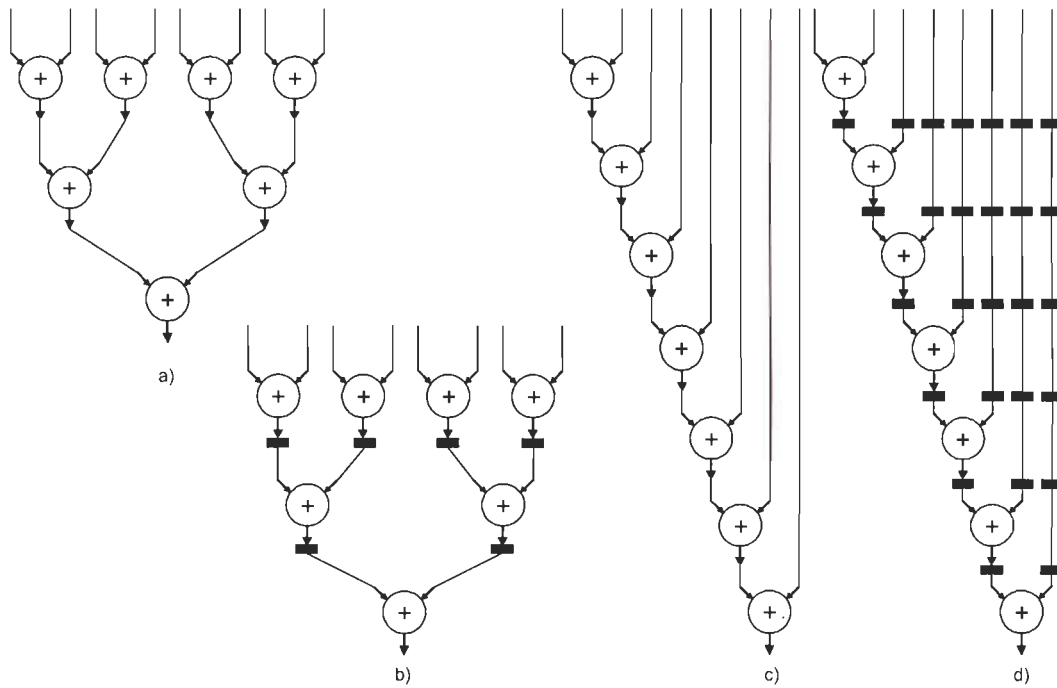


Figure 43 : Circuit de MRC avec les additionneurs à deux entrées : a) Structure parallèle, b) Structure parallèle et pipeline c) Structure série et d) Structure série et pipeline.

IV.3. Résultats de simulations et d'implémentation

Les résultats multi-débit consistent en des simulations de l'implémentation de l'algorithme CF-MUD étendu pour des systèmes multi-débit. Dans ces résultats, sauf indication contraire, nous avons considéré les scénarios de communications WCDMA ci-dessous pour nos travaux d'expérimentation :

- Le canal est variant en temps de type Rayleigh avec 3 trajets de données complexes.
- Le délai relatif maximum entre usagers et le délai relatif maximum entre trajets d'un même usage est respectivement de 30 brides (0,78 ms) et 10 brides (0.26 ms).
- Les paramètres du canal varient en temps, et changent à chaque *slot*.

En plus, pour diminuer le temps des simulations, nous avons développé les modèles (*Rake*, soft MPIC et Signature adaptative) en C/C++ et simulé dans l'environnement de MATLAB®.

Pour des preuves de concept de notre méthode multi-débit, nous considérons le premier bloc du récepteur CF-MUD, c.à.d. le bloc Signature. Nous l'appelons alors la Signature adaptative multi-débit. La Signature adaptative seule offre des performances acceptables pour les systèmes WCDMA. Nous avons considéré 200 itérations pour la phase d'adaptation dans l'ensemble des simulations de cette section, sauf indication contraire. En fait, la figure 44 présente les performances de la Signature adaptative (Sign-MUD) par rapport au récepteur *Rake* (*Rake*) et récepteur *soft* MPIC (4-stage Soft MPIC) à 4 étages incluant le *Rake* au premier étage.

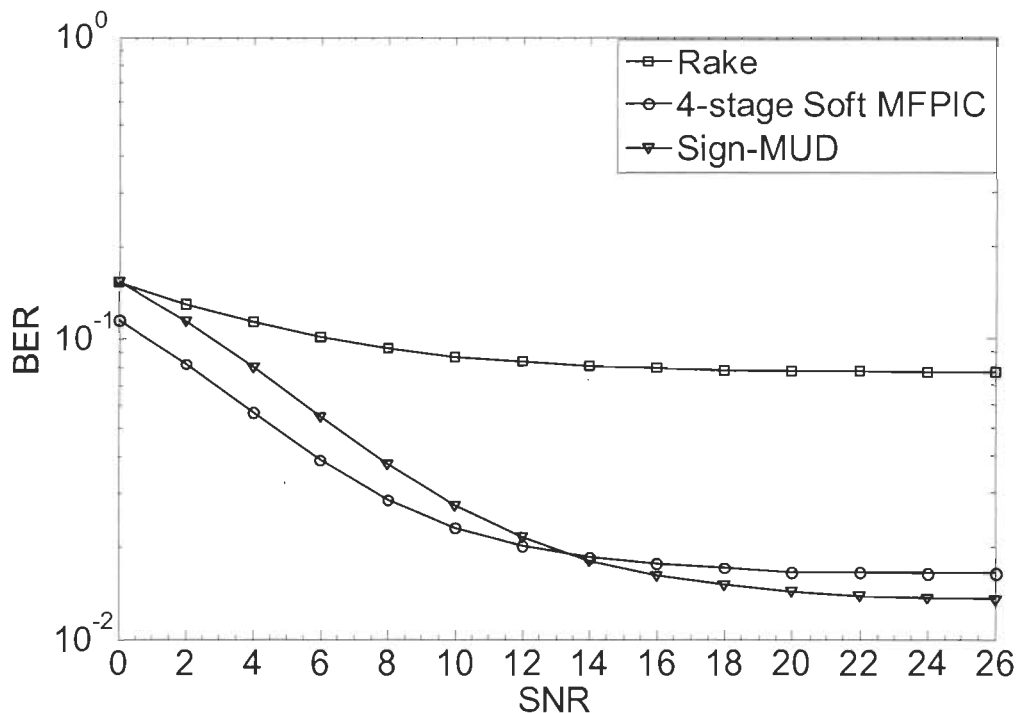


Figure 44 : Performances de la Signature adaptative par rapport au *Rake* et soft MPIC à 4-étage dans un système mono-débit de 16 usagers et de 64 kb/s.

Ainsi, pour montrer des performances de la Signature adaptative vis-à-vis des récepteurs connus, nous avons considéré les scénarios d'un système mono-débit. Par conséquent, la figure 44 montre les performances en termes de BER de la Signature adaptative par rapport à celles du récepteur *Rake* et du récepteur *soft* MPIC à 4 étages. Le système mono-débit considéré étant celui de 64 kb/s avec 16 usagers. À un BER de 5%, nous observons une perte de moins de 2 dB comparativement au détecteur Soft-MPIC (4 étages) et un croisement des courbes à partir d'un

SNR de 14 dB. Cependant, il est clairement démontré dans cette figure que les performances de la Signature adaptative sont au-delà de celles du *Rake*.

Les performances algorithmiques de deux schémas multi-débit sont reportées dans la figure 45. Dans ces expérimentations, nous avons considéré un système à double débit ($G = 2$) avec 4 usagers à 64 kb/s (facteur OVSF=16) et 4 usagers à 144 kb/s (facteur OVSF=8). On constate qu'à partir des résultats de cette figure le schéma LRD offre des performances meilleures que le schéma HRD en termes de BER. Le décalage des performances accroît en augmentant le rapport signal sur bruit (SNR). Les méthodes multi-débit croisent 5% BER à 6.2dB et à 7.5dB pour la méthode LRD et HRD respectivement, donnant un gain de 1.3dB pour la méthode LRD.

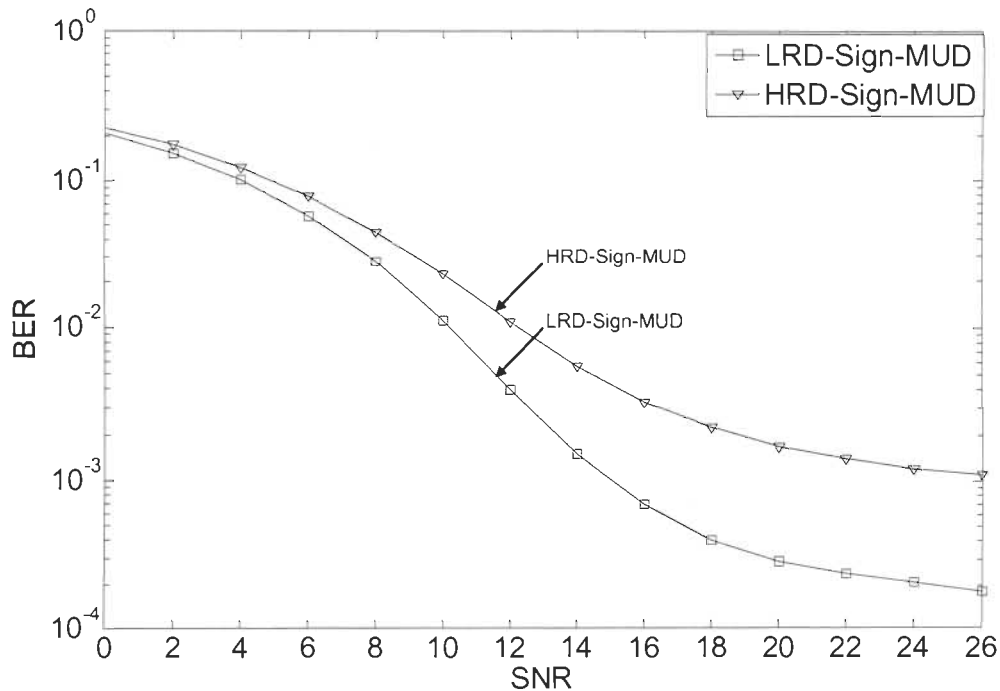


Figure 45 : Performances du schéma LRD-Sign-MUD vs HRD-Sign-MUD avec un système à double débit ($G=2$) : 4 usagers à 64 kb/s et 4 usagers à 144 kb/s.

Les figures 46 et 47 démontrent les performances en virgule fixe du schéma LRD-Sign-MUD (schéma multi-débit de la Signature adaptative basée sur la méthode LRD) et HRD-Sign-MUD (schéma multi-débit de la Signature adaptative basée sur la méthode HRD) respectivement. Ces évaluations ont été effectuées pour un système à double débit ($G = 2$) avec

4 usagers à 64 kb/s (facteur OVSF=16) et 4 usagers à 144 kb/s (facteur OVSF=8). Nous avons considéré 200 itérations pour l'adaptation des coefficients. À partir de ces figures, nous trouvons que la longueur de 13-bit avec format(13,2,11) mène à des performances satisfaisantes. Ce format (13,2,11) dénote la longueur de 13 bits avec 2 bits entiers et 11 bits fractionnaires. Évidemment, la longueur des nombres fixes affecte les performances d'une part. D'autre part, on peut également trouver que le format des nombres fixes affecte davantage des performances. En effet, le format (16,1,15) offre des performances pires que le format(13,2,11). Cette perte de performances du format (16,1,15) par rapport de celles du format (13,2,11) est due au fait que les données transmises sont du BPSK. Or, on peut facilement démontrer que les valeurs intermédiaires des données estimées sont toujours limitées par la borne inférieure de -2 et la borne supérieure de $+2$.

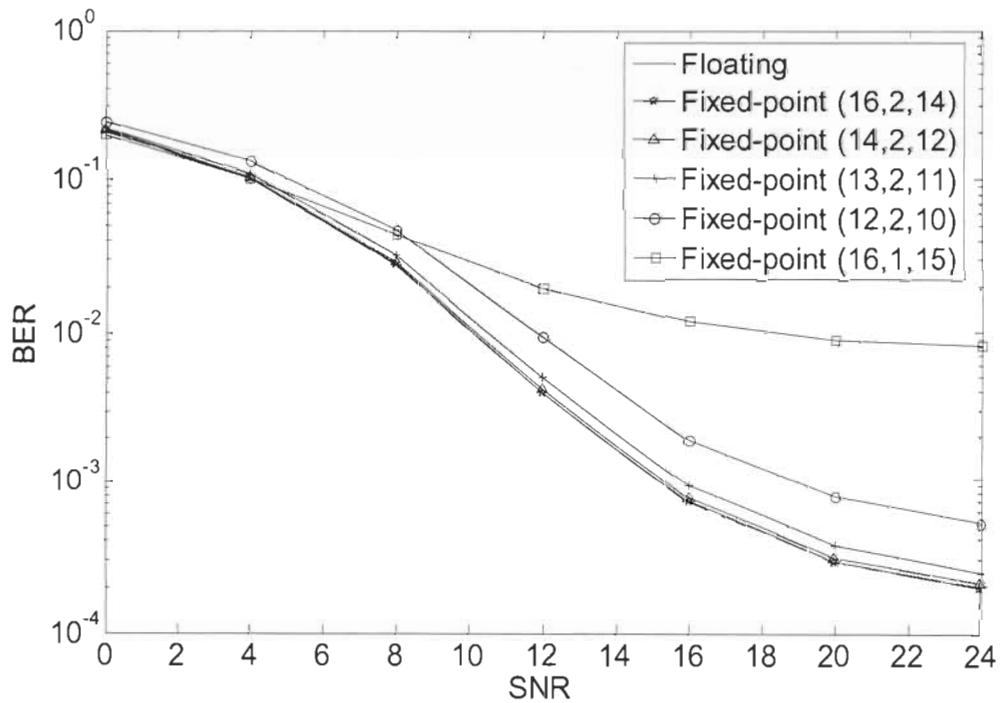


Figure 46 : Performances virgule fixe du schéma LRD-Sign-MUD pour un système à double débit ($G=2$) : 4 usagers à 64 kb/s et 4 usagers à 144 kb/s.

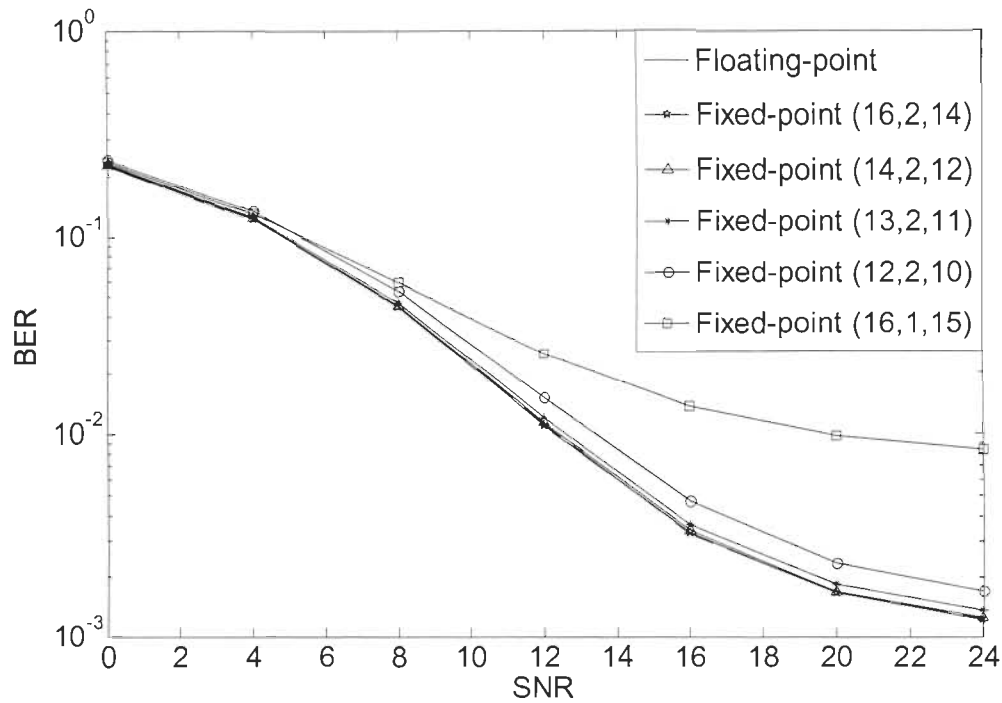


Figure 47 : Performances virgule fixe du schéma HRD-Sign-MUD pour un système à double débit : 4 usagers à 64 kb/s et 4 usagers à 144 kb/s.

À partir de cette figure, nous trouvons que la longueur de 13-bit avec format(13,2,11) satisfait des performances. Aussi, la longueur et le format des nombres fixes affecte des performances. En effet, le format (16,1,15) offre des performances pires que donne le format(13,2,11). Cette perte de performances du format (16,1,15) par rapport de celles du format (13,2,11) est due par le fait que les données transmises sont de BPSK. Or, on peut facilement démontrer que les valeurs intermédiaires des données estimées sont toujours limitées par la borne inférieure de -2 et la borne supérieure de $+2$.

Le tableau 14 montre la complexité de la Sign-MUD en nombre d'opérations arithmétiques et d'accès mémoire. Cette analyse est importante et sert à l'évaluation des performances matérielles aussi bien pour le développement d'une architecture spécialisée que sur un processeur général de traitement des signaux (DSP). En fait, pour un processeur DSP donné, cette analyse permet de savoir s'il peut répondre ou pas en temps réel. Dans des contextes multi-débit, cette analyse nous aide particulièrement à estimer les performances de nos schémas multi-

débit étant donné que nous dérivons un système multi-débit basé sur un système mono-débit. Dans ce tableau, N_f dénote la longueur des filtres de la Signature adaptative ; N_{adapt} dénote le nombre d'itérations d'adaptation ; N_{detect} dénote le nombre de symboles à détecter et K dénote le nombre d'utilisateurs.

Tableau 14 : Complexité matérielle des schémas multi-débit de Sign-MUD.

Opération	Complexité
Multiplication	$\left[(6N_f + 1)N_{adapt} + 2N_f N_{detect} \right] K$
Addition	$\left[(8N_f + 1)N_{adapt} + 2N_f N_{detect} \right] K$
Accès mémoire	$\left[(14N_f + 2)N_{adapt} + 8N_f N_{detect} \right] K$

La figure 48 montre la complexité matérielle de deux méthodes multi-débit d'un système à double débit ($G=2$) en fonction du nombre d'utilisateurs réels simultanés, K , et du rapport entre le débit élevé et le débit faible défini par le rapport des OVFS, M . Dans cette évaluation, nous avons considéré le même nombre d'utilisateurs réels à débit élevé et à débit faible. Par exemple, sur cette figure, pour un nombre d'utilisateurs réels observé de 8, il y a 4 utilisateurs réels à débit faible et 4 utilisateurs réels à débit élevé.

Pour évaluer la complexité matérielle en fonction du rapport M (rapport du débit élevé sur le débit faible), nous avons fixé le débit faible à 12.2 kb/s ($OVFS = 64$) et fait varier le débit élevé à 12.2 kb/s, 64 kb/s et 144 kb/s ($OVFS=64, 16$ et 8 respectivement). Dans cette étude de complexité, nous avons considéré 100 itérations pour réaliser la phase d'adaptation.

On observe que le nombre d'opérations – addition, multiplication et accès mémoire – croît excessivement en augmentant le rapport M pour la méthode RLD. Par contre, pour la méthode HRD, le nombre d'opérations croît très légèrement en augmentant le rapport M . Aussi, on remarque que le nombre d'opérations est quasiment constant en fonction de M . Cependant, en valeurs numériques, cela augmente très légèrement.

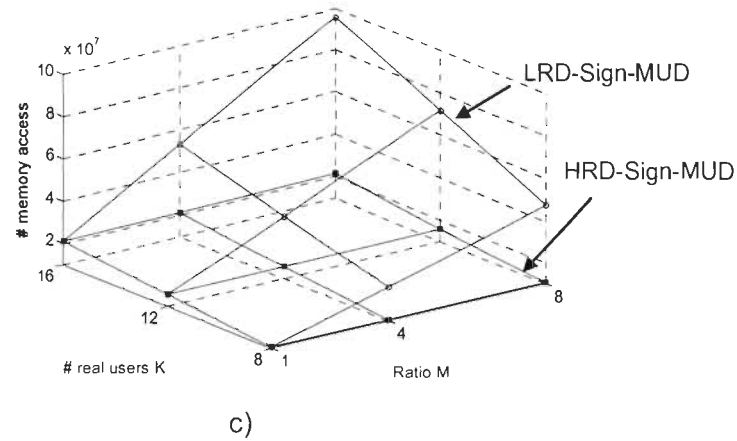
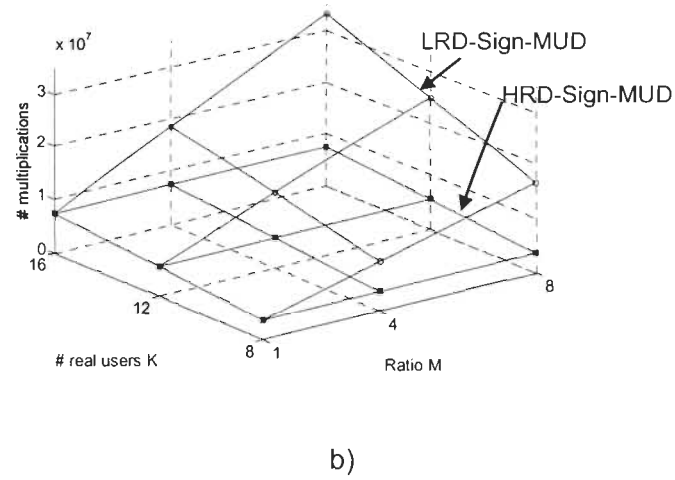
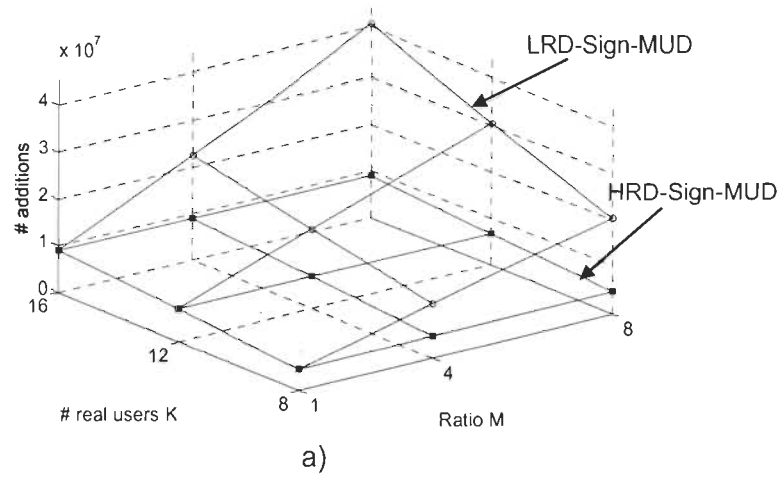


Figure 48 : Complexité matérielle du schéma LRD-Sign-MUD vs. HRD-Sign-MUD dans un système à double débit ($G=2$) en termes de nombre : a) d'additions, b) de multiplications, et c) d'accès mémoire.

Le tableau 15 montre le nombre maximum d'utilisateurs équivalents d'un système multi-débit sur quelques composants de la famille Virtex II Pro. Dans ce tableau, nous considérons le nombre d'utilisateurs équivalents¹³ du système mono-débit équivalent dérivé par une méthode multi-débit que ce soit LRD-Sign-MUD ou HRD-Sign-MUD, le nombre d'utilisateur équivalent est le même. Les conditions d'évanouissement considérées consistent en l'environnement semi-stationnaire et stationnaire défini par l'opération de la phase d'adaptation effectuée à 10ms et 40ms respectivement. Les débits considérés sont ceux de données de 64, 144 et 384 kb/s. Ainsi, ces résultats de ce tableau nous permettent d'estimer le nombre d'utilisateurs équivalent d'une méthode multi-débit qui peut être intégré sur un composant FPGA. Ce tableau montre un nombre important d'utilisateurs équivalent sur un cible FPGA.

Tableau 15: Nombre maximum d'utilisateurs équivalent (K^{MAX}) des systèmes multi-débit sur les composants de la famille Virtex II Pro pour la méthode LRD-Sign-MUD ou HRD-Sign-MUD.

Device \ Rate	Semi-stationnaire			Stationnaire		
	64 kb/s	144 kb/s	384 kb/s	64 kb/s	144 kb/s	384 kb/s
VP4	14	14	14	7	7	7
VP7	22	22	22	14	14	14
VP20	44	44	44	22	22	22
VP30	68	68	68	34	34	34

Nous avons aussi réalisé l'étude de la complexité pour la méthode CF-MUD dans un système multi-débit. Les tableaux 16 et 17 montrent les résultats de cette étude du récepteur CF-MUD multi-débit basé sur la méthode LRD (LRD-CF-MUD) et HRD (HRD-CF-MUD) respectivement. Nous avons considéré le récepteur CF-MUD multi-débit complet incluant le bloc Signature et le bloc Détection pour les systèmes à trois débits de données soient de 64 kb/s, 144 kb/s et 384 kb/s (facteurs OVFSF correspondants sont 16, 8, 4). Nous avons considéré 100 et $100(256/N_c^{eq})$ itérations d'adaptation pour le bloc Signature et le bloc Détection respectivement, où N_c^{eq} dénote le facteur OVFSF du système équivalent. Le débit équivalent du détecteur LRD-CF-MUD est à 64 kb/s et celui du détecteur HRD-CF-MUD est à 384 kb/s. Le nombre d'utilisateurs réels de chaque groupe de débit est inférieur au facteur OVFSF correspondant. On

¹³ Usager équivalent dans le système mono-débit dérivé

trouve que la méthode produit plus d'utilisateurs équivalents et alloue plus de ressources matérielles. Ces résultats sont conformes avec les études théoriques réalisées à la figure 48 et au tableau 14.

Tableau 16 : Nombre d'utilisateurs implémenté sur les composants de la famille Virtex II Pro et le taux d'utilisation respectif du schéma LRD du récepteur CF-MUD multi-débit.

Rate Device	Stationnaire					Semi-stationnaire				
	64 kb/s	144 kb/s	384 kb/s	LRD		64 kb/s	144 kb/s	384 kb/s	LRD	
				# usagers équivalents	%				# usagers équivalents	%
VP2	4	1	1	10	97	4	1	0	6	88
VP4	14	1	1	20	100	8	1	1	14	100
VP7	16	4	1	28	95	12	1	1	18	95
VP20	16	8	4	48	98	16	4	1	28	99
VP30	16	8	4	48	70	16	6	1	32	97
VP40	16	8	4	48	58	16	8	1	36	99

Tableau 17 : Nombre d'utilisateurs implémenté sur les composants de la famille Virtex II Pro et le taux d'utilisation respectif du schéma HRD du récepteur CF-MUD multi-débit.

Rate Device	Stationnaire					Semi-stationnaire				
	64 kb/s	144 kb/s	384 kb/s	HRD		64 kb/s	144 kb/s	384 kb/s	HRD	
				# usagers équivalents	%				# usagers équivalents	%
VP2	1	1	4	6	88	1	1	2	4	89
VP4	2	8	4	14	100	1	1	4	6	100
VP7	6	8	4	18	95	2	2	4	8	97
VP20	16	8	4	28	57	4	8	4	16	97
VP30	16	8	4	28	39	4	8	4	16	94
VP40	16	8	4	28	33	4	8	4	16	67

IV.4. Conclusions

Dans ce chapitre, le noyau de l'algorithme CF-MUD est utilisé pour concevoir et implémenter un récepteur multi-débit. Pour des raisons de simplification, nous avons considéré la Signature seul (Sign-MUD) – le premier bloc constituant du récepteur CF-MUD pour des systèmes multi-débit. Nous considérons deux schémas multi-débit couramment utilisés dans la littérature (LRD et

HRD). Bien que notre approche qui génère les usagers virtuels soit différente de celle de la littérature, les simulations ont démontré que nos résultats coïncident avec ceux des travaux existants. En effet, la méthode LRD offre des meilleures performances en termes de taux d'erreur, mais introduit davantage de surplus en ressources matérielles.

CHAPITRE V

CONTRIBUTION À LA MÉTHODOLOGIE DE CONCEPTION

En premier lieu, nous allons présenter le flot de conception général. Ensuite, nous allons présenter notre contribution à la méthodologie de conception concernant l'approche conventionnelle et l'approche unifiée. La dernière consiste à utiliser un seul langage et dans un seul environnement de conception. Aussi, le développement des bibliothèques spécialisées en virgule fixe sera présenté.

V.1. Flot de conception général

La figure 49 présente le flot de conception général d'une application cible sur FPGA. Ce flot de conception peut être vu comme une suite de la méthodologie proposée dans [66]. Il consiste relativement en quatre étapes principales. Le niveau d'abstraction de chaque étape est indiqué par un ton de gris. En fait, plus le niveau de gris est foncé, plus le niveau d'abstraction est bas. Ainsi, étant donné notre application, les quatre étapes se réalisent comme suit :

- Étape1 – « Développement de l'algorithme » : L'algorithme d'une application est développé incluant sa modélisation et évaluation selon des spécifications algorithmiques préalablement posés et basées sur des contraintes et critères d'évaluations algorithmiques propres à l'application. Une fois que l'algorithme est fonctionnel, ses résultats de simulations constituent une référence pour valider l'étape suivante –

élaboration de l'algorithme en vue d'une implémentation VLSI. Cette étape est communément réalisée en virgule flottante dans l'environnement MATLAB® ou en langage C.

- Étape 2 – « Élaboration de l'algorithme en vue d'implémentation VLSI » : L'algorithme fonctionnel d'une application est élaboré en vue d'implémentation VLSI sur une cible technologique (DSP, AS-DSP, FPGA ou ASIC). À cette étape, les spécifications architecturales doivent être définies en termes de contraintes et critères architecturaux propres à l'application. Pour ce faire, en premier lieu l'algorithme est modélisé et simulé en virgule fixe. La version en virgule fixe est fonctionnelle quand elle respecte les contraintes et critères de l'étape 1. Dans cette étape, l'architecture matérielle est conçue de manière conceptuelle (comportementale). Aussi, nous établirons les spécifications architecturales correspondant aux spécifications algorithmiques de la version en virgule fixe de l'algorithme.
- Étape 3 – « Développement de l'architecture matérielle » : L'architecture matérielle est modélisée avec un langage (VHDL, SystemC, C/C++) et simulée, puis synthétisée dans une technologie cible. Ainsi, les opérations de cette étape consistent en la modélisation, la simulation et la synthèse.
- Étape 4 – Élaboration de l'architecture matérielle développée : Cette étape consiste à établir les opérations séquentielles – simulation temporelle, placement & routage (PAR) et simulation après PAR.

Les résultats quantitatifs obtenus à chacune de ces étapes doivent satisfaire les spécifications préalablement posées avant de passer à l'étape suivante. Dépendamment de la méthodologie de conception, les langages et les outils de conception utilisés sont différents. Nous allons les exposer dans la suite de ce chapitre.

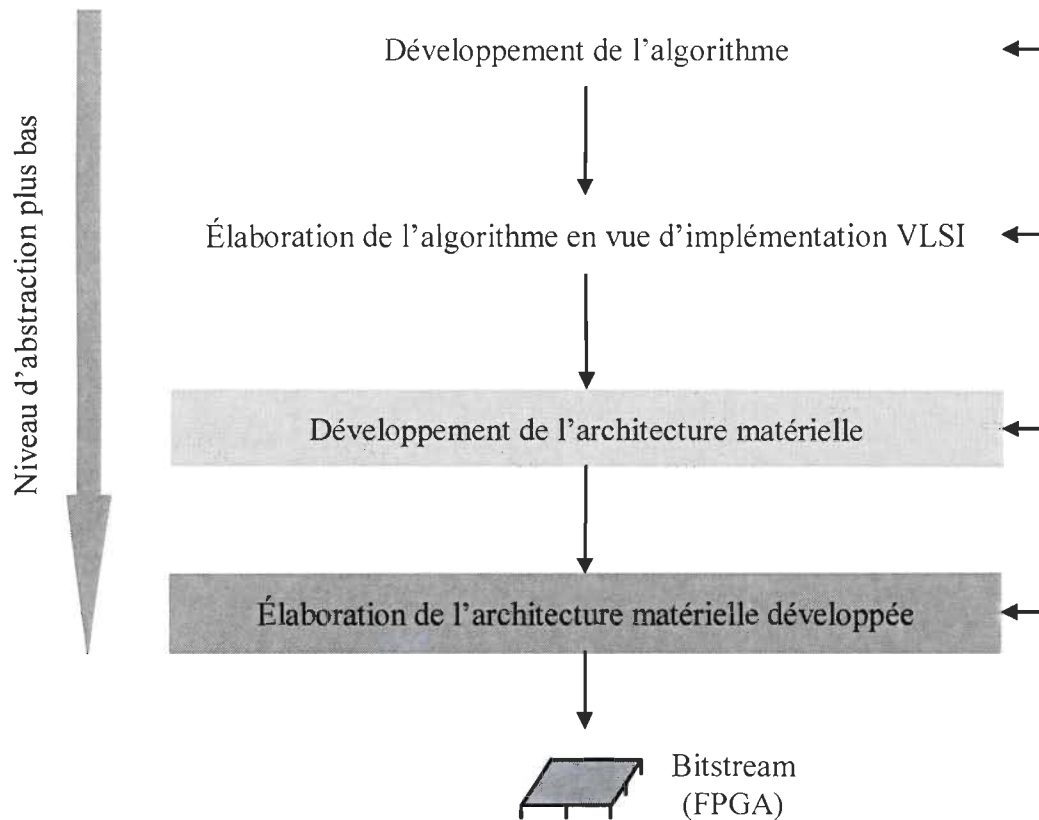


Figure 49 : Flot de conception général cible sur FPGA.

V.2. Approche conventionnelle

La figure 50 présente le flot de conception conventionnel [59] sur la cible FPGA. Nous avons utilisé ce flot durant le développement de nos architectures sur FPGA. Nous avons amélioré le flot de conception en ajoutant la possibilité d'estimer les ressources matérielles à partir d'une spécification architecturale. Ainsi, les opérations de ce flot sont expliquées ci-dessous :

« Simulation en virgule fixe » consiste en plusieurs simulations d'un algorithme d'intérêt selon des scénarios suffisamment larges pour en tirer des conclusions. Cela nous permet d'avoir la version en virgule fixe de l'algorithme à implémenter tout en respectant les performances algorithmiques attendues. Comme une implémentation en virgule flottante est souvent impraticable en termes de coût, principalement dû à la surface et la consommation, cette étape est critique. Les simulations s'effectuent généralement à l'aide d'une librairie virgule fixe dans

l'environnement MATLAB® [RT1], ou à l'aide des types de données Algorithmic C™ de Mentor Graphics [70]. Dans cette étape, un modèle à virgule fixe en MATLAB® ou C/C++ de l'algorithme étudié est réalisé et intégré dans l'environnement de simulation MATLAB®. Finalement, cette étape nous permet d'avoir des performances avec précision non seulement au niveau système mais aussi au niveau bit incluant l'effet de la quantification.

« Spécification algorithmique » consiste en les spécifications d'un algorithme d'intérêt. Pour le CF-MUD, par exemple, les spécifications algorithmiques consistent¹⁴ en le facteur d'étalement OVFSF, le nombre de coefficients par filtre N_f , le pas d'adaptation μ , le nombre d'itérations de chaque bloc, le nombre d'utilisateurs K , le nombre d'utilisateurs traités par filtre, le multiplexage temporel, le parallélisme.

« Spécification architecturale » permet d'exposer les spécifications architecturales nécessaires pour implémenter sur une cible. Cette étape est obtenue après avoir obtenu la version en virgule fixe de l'algorithme d'intérêt tout en tenant compte de la particularité de la cible technologique. Il s'agit du développement conceptuel de l'architecture.

« Spécification & caractérisation » permet de spécifier et caractériser les éléments de base de la librairie. La librairie se compose des éléments pré-développés et pré-vérifiés tels que les multiplieurs, les additionneurs, les diviseurs, et les autres. Ces éléments de librairie sont spécifiques et exploitent la particularité de la cible technologique. L'utilisation d'une telle librairie permet de réduire le temps de développement considérant que ces éléments de base ont déjà été vérifiés auparavant.

« Estimation de ressources » permet d'estimer les ressources matérielles d'un design conceptuel de l'architecture matérielle sans besoin de synthétiser son modèle HDL¹⁵. Cette étape est extrêmement importante car elle permet de concevoir conceptuellement l'architecture d'une spécification architecturale tout en respectant les ressources exigées par cette dernière. Puis, elle fournit des informations au retour à l'étape « Spécification architecturale » pour valider ou

¹⁴ La définition de ces variables sont donnée dans la section III.1

¹⁵ Dans ce travail à la section V.2, l'estimation de ressources matérielles d'une spécification architecturale est effectué avant même la modéliser en HDL.

réviser le design conceptuel. Dans le cadre de ce travail de thèse, nous avons développé une méthode d'estimation des ressources matérielles basée sur un modèle de programmation non-linéaire en nombres entiers. Cette méthodologie est applicable à un large éventail de classes d'algorithmes qui sont basés sur le calcul récursif et/ou le calcul matriciel [RT4].

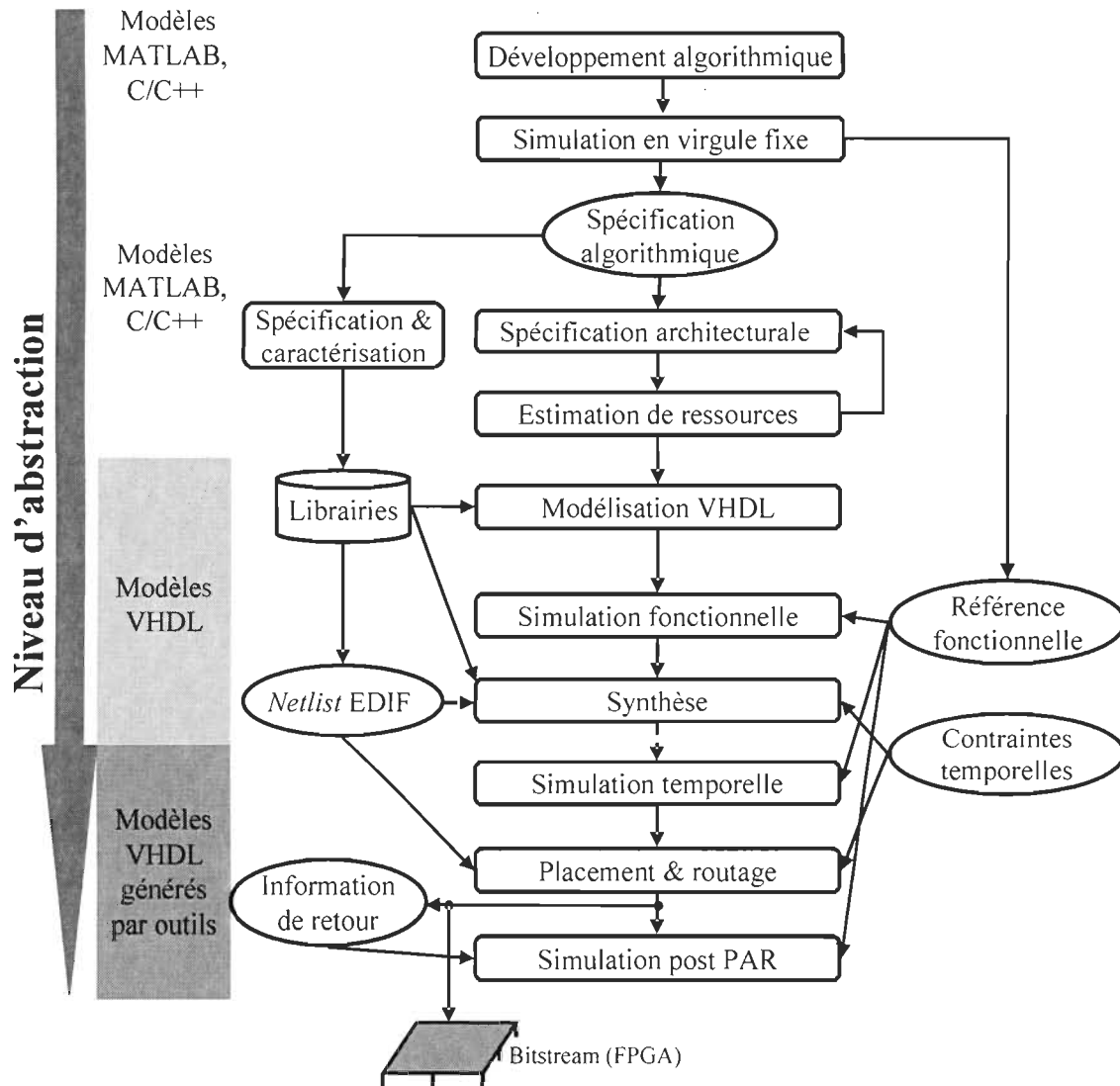


Figure 50 : Approche conventionnelle d'un flot de conception.

« Modélisation VHDL » permet de modéliser l'architecture matérielle une fois que cette dernière a été validée. Cette étape représente le travail le plus important dans le flot de conception. Le développement de la partie fonctionnelle de l'architecture est accéléré par

l'utilisation de la librairie des éléments de base pré-vérifiés. Toutefois, la modélisation de la partie contrôle de l'architecture reste non-triviale et représente un défi majeur du développement matériel.

Simulation fonctionnelle permet de vérifier le fonctionnement de l'architecture sans informations temporelles de la cible technologique. Cette étape est nécessaire car elle vérifie le modèle VHDL développé indépendamment de la cible technologique.

Les autres opérations du flot sont communes aussi bien pour l'approche conventionnelle que pour l'approche unifiée. En effet, la « Synthèse » permet de synthétiser le modèle VHDL sur la cible – composants FPGA Virtex II et Virtex II Pro dans nos travaux. Aussi, la vérification statique a lieu dans cette étape. La « Simulation temporelle » permet de simuler le modèle synthétisé avec informations temporelles de la cible technologique données par l'outil de synthèse. Le « Placement & routage » permet d'émettre la liste des connexions (*netlist*) du circuit. Bien que les outils de placement & routage, par exemple ISE Foundation™ de Xilinx®, soient capables de considérer directement un modèle comportemental en VHDL, on utilise la *netlist* en format standard EDIF. La « Simulation post PAR » consiste en la simulation temporelle avec informations temporelles données par l'outil de PAR. Ces informations temporelles sont plus précises que celles données par l'outil de synthèse. La simulation de cette étape s'effectue alors à l'aide d'un modèle structurel VHDL des portes logiques de la cible, et d'un fichier des informations temporelles. Ces derniers sont donnés par l'outil de placement & routage.

Un exemple d'estimation de ressources matérielles :

Dans la suite de cette section, nous verrons un exemple d'estimation de ressources matérielles d'une spécification architecturale d'un récepteur telle que décrite dans la section III.2 du chapitre III sur les familles FPGA Virtex-II et Virtex-II Pro. La tâche consiste à établir le modèle de programmation incluant les fonctions objectives et jeu de variables, et à implémenter ce modèle établi. Aussi, nous présenterons brièvement le modèle général de programmation non linéaire en nombres entiers.

1. Modèle général de programmation non linéaire en nombres entiers

Un modèle général de programmation non linéaire en nombres entiers est décrit comme suit [81] :

$$\begin{aligned} \max \{z(x)\} \\ g_i(x) \leq 0, i = 1, \dots, m \end{aligned} \quad (74)$$

où z est la fonction objective (ou les fonctions objectives); g_i sont les fonctions de contraintes; où g_i sont les non linéaires de valeurs réelles.

2. Fonction objective et jeu de variables

L'objectif de ce modèle de programmation consiste à chercher la valeur optimale du nombre d'utilisateurs K sur FPGA cibles. Étant donné la spécification architecturale du détecteur comme décrite dans la section III.2 et la structure matérielle des FPGA cibles, la fonction objective est établie comme suit (cf. [RT4])¹⁶ :

$$K \leq \frac{1024}{3T_{MUX2}} \quad (75)$$

$$K \leq \frac{t_D}{9N_2T_{MUX2}} - 1 \quad (76)$$

$$K \leq \frac{1}{6} \left(\frac{t_D}{\frac{38400}{N_c} T_{MUX2}} - 5 \right) \quad (77)$$

$$K \leq \frac{N_m T_{MUX1} T_{MUX2}}{4(T_{MUX1} + T_{MUX2})} \quad (78)$$

¹⁶ Ce développement a été effectué sur les composants Virtex 2 et Virtex 2 Pro. 1024 est la taille d'un bloc RAM.

En respectant les contraintes suivantes :

- T_{mux1} est un entier et doit satisfaire les contraintes ci-dessous :

$$T_{MUX1} \leq 4 \quad (79)$$

$$T_{MUX1} \leq \frac{t_A}{N_1 \frac{256}{N_c} (3N_c + 9)} \quad (80)$$

$$T_{MUX1} \leq \frac{t_A}{\frac{38400}{N_c} (2N_c + 5)} \quad (81)$$

- Les variables N_1, N_2, N_c, t_A, t_D sont constantes.

où T_{MUX1} et T_{MUX2} dénotent le multiplexage temporel de l'architecture matérielle du bloc Signature et Détection comme définis dans la section III.2 (tableau 4) ;

t_A et t_D dénotent le temps d'adaptation et de détection comme définis dans la section III.8 ;

N_1, N_2 dénotent le nombre de PE dans le bloc Signature et Détection comme définis dans la section III.8 ;

N_m dénote le nombre de multiplieurs câblés et BRAM disponibles sur ces FPGA cibles.

Le jeu de variables est assez limité étant donné que les variables N_1, N_2, N_c, t_A et t_D sont constantes. Ainsi, il contient T_{MUX1} et T_{MUX2} respectant les fonctions objectives ci-haut.

3. Implémentation du modèle

Le modèle mathématique formulé dans 2) a été résolu par la méthode « borne et branche » [81]. Un programme a été développé en MATLAB pour résoudre ce modèle. Ainsi, on peut estimer les ressources matérielles nécessaires pour cette spécification architecturale.

V.3. Approche unifiée

V.3.1. Motivations d'une approche unifiée

Le flot conventionnel de conception présenté à la figure 50 présente des limitations majeures, entre autres, concernant le temps de développement et de vérification. En effet, la traduction des modèles entre plusieurs langages et environnements est sujette aux erreurs et consomme du temps. Aussi, l'approche conventionnelle exige que les concepteurs maîtrisent plusieurs langages et environnements de conception. En plus, le développement architectural peut inclure une partie matérielle et une partie logicielle. Dans notre contexte, le développement algorithmique et architectural, l'approche conventionnelle ne présente pas de vérification interactive. En fait, la vérification du développement architectural se réfère à une référence fonctionnelle de l'algorithme. Cette référence fonctionnelle exige une quantité énorme de données résultantes des simulations algorithmiques, notamment avec un très long vecteur de test.

Ainsi, l'approche unifiée transitant moins entres langages et environnements de conception, voire un langage et un environnement, s'avère grandement avantageuse. D'ailleurs, aujourd'hui le langage MATLAB® et son environnement sont choisis pour conception et développement d'un vaste éventail d'applications. Dans notre contexte, le développement algorithmique s'effectue principalement dans l'environnement MATLAB® d'une part. D'autre part, plusieurs travaux d'implémentation architecturale sont déjà réalisés dans l'environnement de MATLAB®, par exemple [10]. Ceci justifie l'un des sous-objectifs de ce travail consistant à développer une approche unifiée de conception dans l'environnement MATLAB®. Pour ce faire, nous proposons deux méthodologies : l'une consiste à étudier et utiliser le langage SystemC dans le flot de conception, et l'autre méthode consiste à utiliser le langage C++ standard dans le flot de conception à l'aide des types de données Algorithmic C™ de Mentor Graphics.

Le choix de ces langages – SystemC et C++ standard – se base sur la facilité de leur interopérabilité avec l’environnement de MATLAB®. En plus, les outils commerciaux sont déjà disponibles pour synthétiser des modèles en C/C++, par exemple Catapult® Synthesis de Mentor Graphics [71].

Dans la suite de cette section, l’interface de MATLAB® avec d’autres langages, notamment C/C++, est succinctement présentée. Cette présentation vise à offrir une meilleure compréhension du développement des travaux qui suivent dans le reste de ce chapitre. Le lecteur peut consulter les manuels de MATLAB® [69].

En effet, pour accélérer le temps de simulations des noyaux des algorithmes dans l’environnement de simulation, MATLAB® fournit aux concepteurs la méthode Mex-C. Ainsi, une fonction Mex-C se compose de deux parties : une passerelle et une fonction opératoire. La fonction opératoire en C/C++ vise à modéliser le noyau d’une application, alors que la passerelle joue le rôle d’interface de la fonction opératoire avec l’environnement MATLAB®. Aussi, l’environnement MATLAB® peut être facilement appelé dans l’environnement C/C++ à l’aide d’un moteur de MATLAB® [69].

V.3.2. Approche unifiée basée sur SystemC

SystemC est une classe de C++ offrant la capacité de développer des systèmes complexes hybrides incluant matériel et logiciel. Le noyau de SystemC est l’ordonnanceur (simulateur) qui a pour but de déclencher ou de reprendre des processus. Typiquement, la simulation d’un modèle SystemC d’une application consiste à faire marcher son programme exécutable dérivé de celui-ci par le compilateur de C++ [83]. La figure 51 présente le cadre unifié de conception basée sur SystemC. Dans cette figure, plus le ton de gris d’un bloc est foncé, plus le niveau d’abstraction est bas.

Dans le flot conventionnel, les langages HDL (ex : Verilog, VHDL) sont utilisés pour la modélisation matérielle. Dans ce flot unifié, SystemC est utilisé pour la modélisation matérielle. Il s’agit de modèles synthétisables en SystemC (SC).

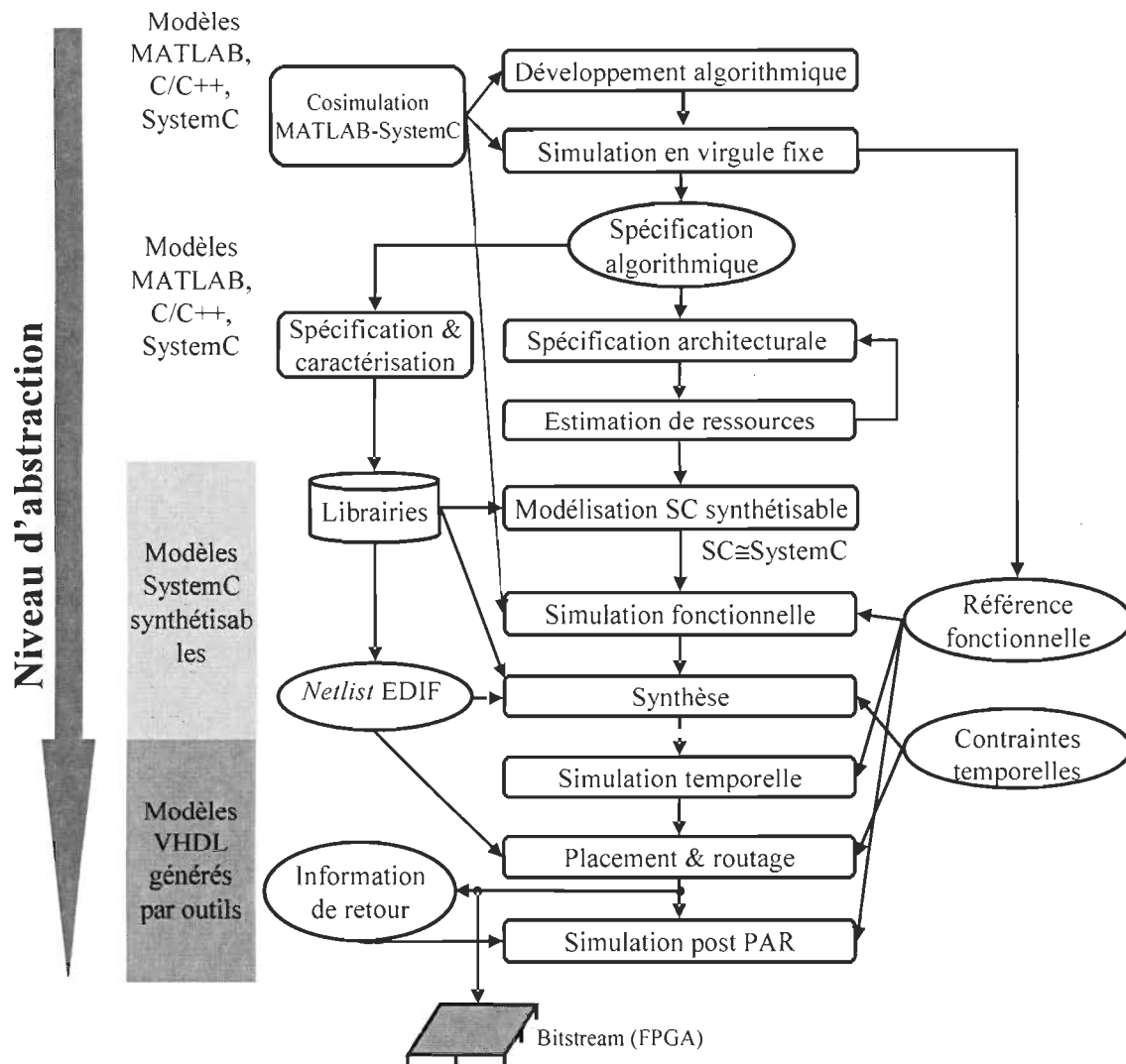


Figure 51 : Approche unifiée basée sur SystemC d'un flot de conception

Dans ce travail on s'intéresse plus particulièrement à une méthode de cosimulation utilisant MATLAB® et SystemC. La problématique majeure se trouve dans l'interopérabilité entre MATLAB® et SystemC. Il existe plusieurs travaux dans la littérature concernant l'interopérabilité entre MATLAB® et autres environnements, et entre SystemC avec d'autres environnements. En effet, un modèle de simulation discrète-continue a été présenté dans [18]. Dans ce modèle, Simulink et SystemC sont respectivement utilisés pour modéliser des composants discrets et des composants continus. Une fonction d'adaptation est enveloppée en fonction-S de Simulink pour l'adaptation de l'interface. Dans ce travail, l'ordonnanceur de

SystemC doit être modifié pour l'adaptation de l'interface entre MATLAB® et SystemC. En plus, les techniques d'interopérabilité reposent sur les outils commerciaux tels que Envoke, COM [34], permettent une cosimulation de SystemC avec d'autres environnements [34]. Aussi, un outil commercial, permettant le lien entre l'outil ModelSim de Mentor Graphics et MATLAB®, est déjà disponible [68]. Toutefois, ces travaux existants exigent une modification non triviale dans le simulateur de SystemC. Dans notre travail, on ne tente pas de modifier l'ordonnanceur de SystemC même, nous tentons plutôt de proposer une méthodologie simple d'interopérabilité qui permet une cosimulation utilisant MATLAB® et SystemC.

V.3.2.A. Cadre de cosimulation MATLAB®-SystemC

Le cadre de cosimulation est basé sur la capacité que :

- le moteur MATLAB® peut être appelé dans l'environnement de SystemC ; et
- l'ordonnanceur SystemC peut être appelé dans l'environnement de MATLAB®.

La figure 52 présente ce principe du cadre de cosimulation utilisant MATLAB® et SystemC. D'ailleurs, le travail se base de la version 2.1 de SystemC [83]. Dans le fond, une simulation de SystemC OSCI (*Open SystemC Initiative*) consiste à faire rouler un programme exécutable dérivé d'un modèle SystemC de l'application. Avant l'arrivée de la version 2.1, l'entrée principale (*main*) de l'application débute avec *sc_main*. À partir de la version 2.1, la *main* peut être directement exportée avec les arguments *argv* et *argc* dans *sc_main*. Cette capacité facilite l'interopérabilité de l'ordonnanceur de SystemC avec autres langages, incluant donc MATLAB®. Toutefois, il existe encore des limitations dans l'actuelle version 2.1 concernant l'ordonnanceur de simulation de SystemC. En fait, la *main* dans un contexte de simulation ne peut être appelée qu'une seule fois. Or, on désire appeler ce contexte de simulations plusieurs fois. La capacité d'appeler un modèle de SystemC plusieurs fois nous permet de synchroniser avec l'environnement MATLAB®.

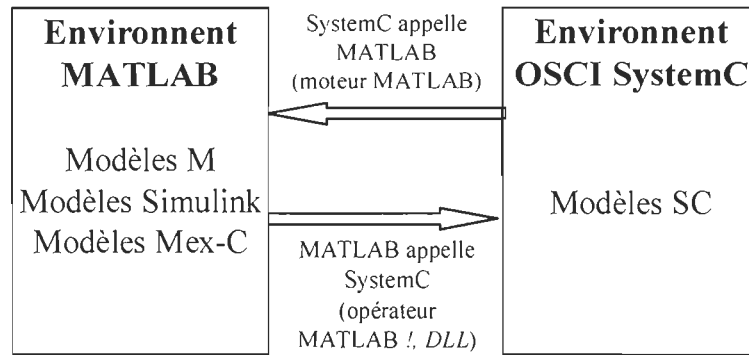


Figure 52 : A cadre de cosimulation utilisant MATLAB® et SystemC.

Le langage MATLAB® offre aux programmeurs la possibilité d'appeler facilement un programme MATLAB® à l'intérieur de l'environnement des autres langages à l'aide du moteur MATLAB®. Évidemment, on peut appeler MATLAB® pour accéder soit aux données, soit aux programmes MATLAB® – modèles M, modèles Mex-C – à l'intérieur d'un noyau de simulation SystemC OSCI. Ainsi, on peut appeler des modèles MATLAB® à l'intérieur de l'environnement SystemC [18].

MATLAB® est doté de l'opérateur « ! » permettant d'appeler un programme exécutable. Évidemment, ce programme exécutable peut être dérivé d'un modèle SystemC d'une application pour la simulation. Par conséquent, en utilisant cette capacité de MATLAB®, on peut appeler des modèles SystemC à l'intérieur de l'environnement de MATLAB®. En plus, une alternative pour appeler l'ordonnanceur SystemC à l'intérieur de l'environnement MATLAB® développé dans ce cadre de cosimulation repose sur des bibliothèques dynamiques (*DLL : Dynamic Link Library*). Cette approche est similaire à la méthode Envoke [34].

Pour des simulations en virgule flottante, MATLAB®, Mex-C et C/C++ partagent le type de données *double* comme en langage C/C++. Toutefois, pour des simulations en virgule fixe, une conversion de données doit être faite. Voici la procédure de conversion :

- 1. Pour des simulations en virgule fixe de SystemC, on utilise la classe *sc_fixed* qui est une représentation binaire définie comme suit :

`sc_fixed<wl, iwl, q_mode, o_mode, n_bits> ;`

où wl dénote la longueur binaire d'un nombre en question ;

iwl dénote la longueur de la partie entière ;

q_mod dénote le mode de quantification ; et

o_mode dénote le mode de débordement.

- 2. Pour des simulations en virgule fixe de MATLAB®, on utilise l'objet *fi*. Une fois qu'une variable d'objet *fi* est déclarée, elle peut être utilisée comme une variable MATLAB® ordinaire pour des opérations arithmétiques. La forme générale d'une variable d'objet se compose d'une valeur de données, d'un objet *fi* et d'un objet *numericity*. Il est invité à consulter [67] pour plus de détails sur la définition de ces objets.

V.3.2.B. Expérimentations

Pour l'expérimentation de cosimulation proposée à la section précédente (cf. V.3.2.A), nous avons utilisé le bloc Signature du récepteur CF-MUD, Sign-MUD, présenté au chapitre III [cf. III.1]. Ainsi, cette expérimentation est présentée à la figure 53. Dans ce travail expérimental, le noyau de Sign-MUD a été modélisé en SystemC (bloc ombré). Cependant, pour des raisons de comparaison, dans les tableaux 19 et 20, le Sign-MUD a été aussi modélisé en MATLAB® et Mex-C. D'autres blocs de traitement incluant l'étalement, transmission de données à travers le canal à données complexes, la modélisation du canal, etc. ont été modélisés en MATLAB®. En plus, le Sign-MUD prend en considération les paramètres connus au préalable pour son traitement comme les délais relatifs entre les usagers et les délais relatifs entre les trajets d'un même usager.

Le Sign-MUD a été modélisé aussi bien en virgule fixe qu'en virgule flottante. Il est partitionné selon ses deux phases : la phase d'adaptation et la phase de détection.

La modélisation en virgule flottante a été réalisée avec les modèles M (*scripts* de MATLAB), Mex-C et SystemC.

La modélisation en virgule fixe de MATLAB® a été réalisée à l'aide de *toolbox* MATLAB® (objet *fi*). La modélisation en virgule fixe SystemC a été réalisée à l'aide de la classe SystemC *sc_fixed*. Les modèles SystemC actuels sont des modèles comportementaux, sans considérer les cycles d'horloge (*untimed modeling*).

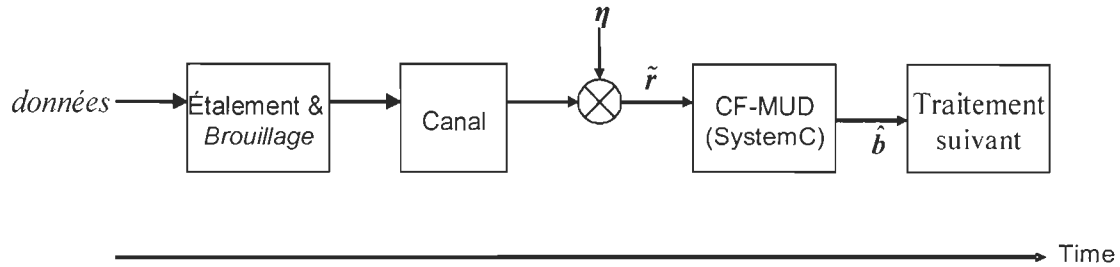


Figure 53 : Cosimulation d'un récepteur CF-MUD utilisant MATLAB® et SystemC.

Les conditions de simulations pour les systèmes WCDMA à débits de données (64 kb/s) reportées au tableau 18.

Tableau 18 : Conditions de simulations du bloc Signature d'un récepteur CF-MUD.

Description	Valeur
Nombre de trames données	66
Facteur OVSF	16
Longueurs des codes courts	256
Nombre d'utilisateurs simultanés	8
Nombre de trajets	3 poids à nombres complexes
Délai max relatif entre trajets d'un même usager	10
Délai max relatif entre usagers	30
Nombre d'itérations pour l'adaptation des filtres	150*(256/OVSF)

Les données transmises consistent en une image découpée en 8 sous-images. Chaque sous-image comprend 66 trames de données et est transmise par un usager distinct. Le nombre d'utilisateurs total est donc 8. La figure 54 présente l'image originale (a), les 8 sous-images bruitées après le canal de transmission (b) et les 8 sous-images reconstruites après le détecteur en virgule fixe. Les simulations ont été effectuées sur une station IBM IntelliStation Z Pro avec un processeur Pentium® à 3.4 GHz et 2.75 GB RAM. Le temps de simulation inclut le temps d'accès des données et le temps de traitement du noyau (Sign-MUD). Pour une comparaison relative, tous les modèles MATLAB® (virgule flottante fixe), Mex-C et SystemC (virgule flottante fixe) ont été modélisés de la même manière, c'est-à-dire avec le même contrôle des boucles, d'accès des données, d'entrée/sorties, etc.

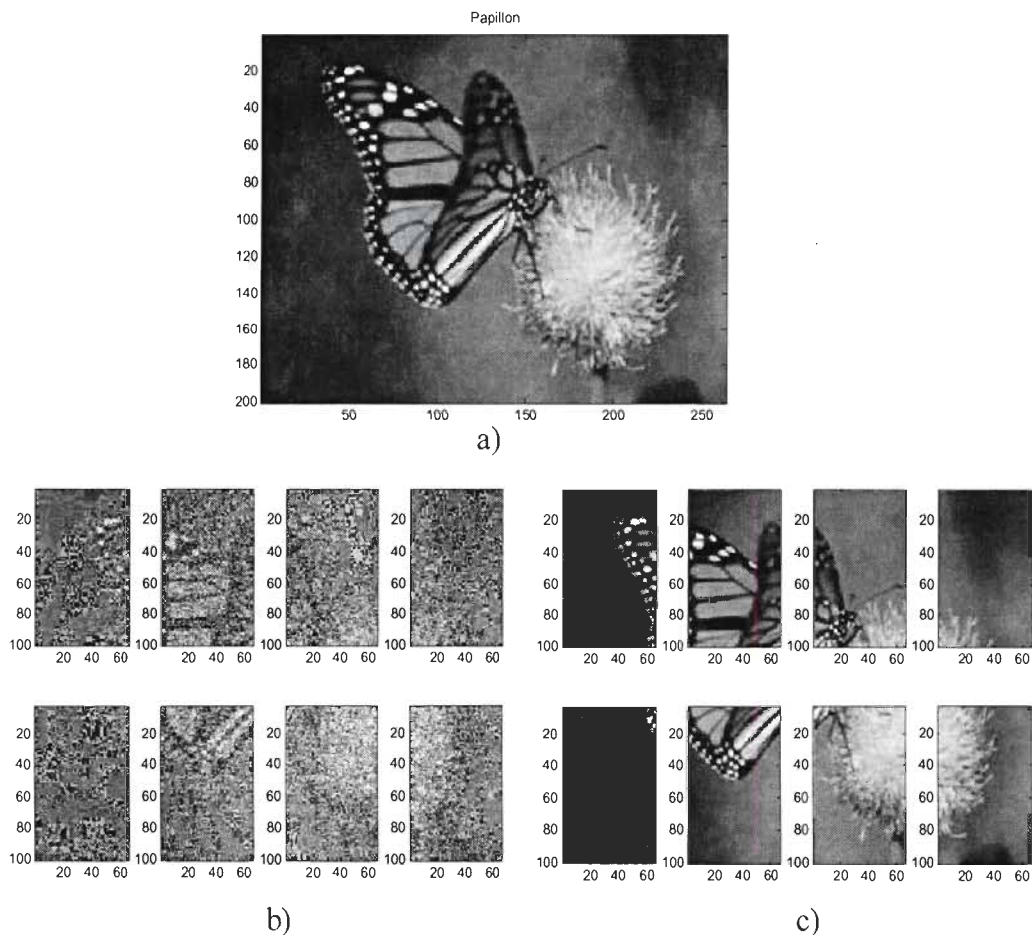


Figure 54 : Données transmises de la figure 54 : a) Image originale, b) 8 sous-images bruitées après le canal de transmission, c) 8 sous-images reconstruites après le récepteur en virgule fixe.

Le tableau 19 démontre les performances de simulations en termes de gain de temps de simulations en virgule flottante et fixe par rapport à la simulation des modèles MATLAB®. On peut noter à partir du tableau 19 que le gain de temps de simulations en virgule flottante des modèles Mex-C et des modèles SystemC est relativement élevé soit de 24 et 9 fois respectivement. Évidemment, le temps de simulation des modèles SystemC comportementaux peut être amélioré davantage en modifiant sa séquence de contrôles. Toutefois, comme constaté auparavant, on voulait, pour des raisons de comparaison, modéliser tous les modèles de la même manière. Concernant les résultats en virgule fixe, il s'agit de nombres fixes au format (16,14) – 16 bits avec 14 bits entiers. Ce tableau montre que le temps de simulation en virgule fixe de SystemC est relativement 2 fois plus rapide qu'en virgule fixe de MATLAB® utilisant l'objet *fi*.

Tableau 19 : Temps de simulation du bloc Signature en virgule flottante et fixe.

Modèles	Virgule flottante		Virgule fixe		Rapport Fixe-Flottante
	Temps (s)	Gain	Temps (s)	Gain	
MATLAB®	74.8	1.00	5 570	1.00	74.5
SystemC	8.73	8.57	3 212	1.74	368
Mex-C	3.08	24.3	-	-	-

V.3.3. Approche unifiée basée sur C/C++ standard

Tenant compte de sa popularité, nous nous intéressons à l'utilisation du langage C++ standard dans le développement matériel. L'outil de synthèse des modèles C++ (Catapult® Synthesis) est commercialement disponible [70]. En plus, en utilisant le langage C++ standard pur, nous n'avons pas besoin d'un simulateur particulier comme SystemC – qui se base sur les particularités de son ordonnanceur. La figure 55 présente le flot unifié de conception basée sur le langage C++ standard.

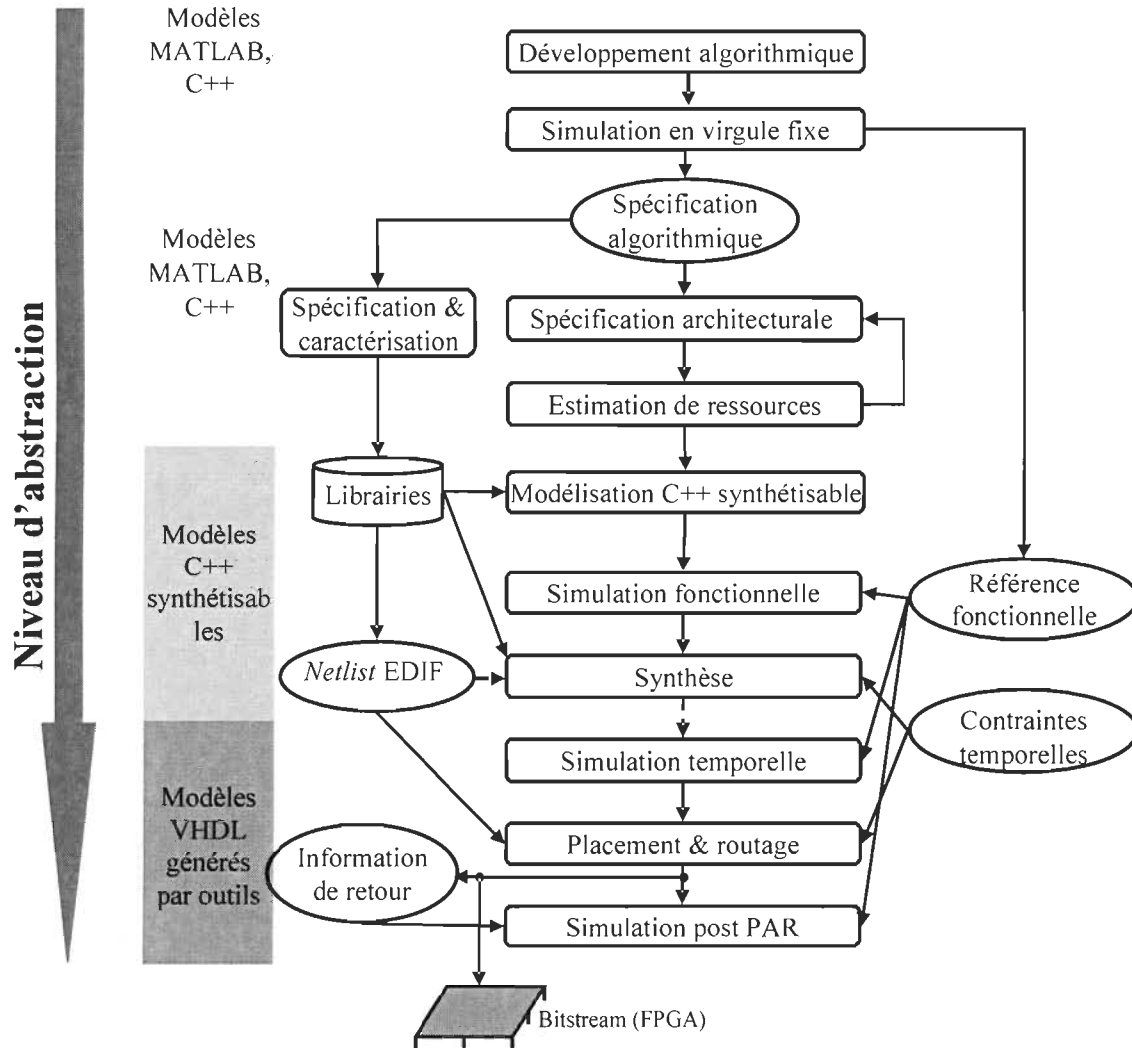


Figure : Approche unifiée basée sur C/C++ standard d'un flot de conception.

Dans ce flot de conception, le langage C++ standard avec contraintes de Catapult® est utilisé pour la modélisation matérielle. Ainsi, des modèles synthétisables en C++ sont réalisés.

Contrairement au flot basé sur SystemC, aucune méthode d'interopérabilité particulière est n'exigée dans ce flot basée sur le C/C++ standard. Toutefois, la passerelle de Mex-C joue le rôle d'adaptation des modèles C++ dans l'environnement. Les expérimentations de ce flot seront exposées dans la section V.4.3 concernant l'utilisation de la classe AC_M.

V.4. Développement des bibliothèques spécialisées en virgule fixe

Le développement d'un paquetage VHDL en virgule fixe et une classe de données virgule fixe MATLAB® est exposé. La présentation des nombres à virgule fixe qui constituent la base de ce développement est également décrite.

V.4.1. Représentation des nombres en virgule fixe

Un nombre entier peut être représenté comme suit :

$$b = b_{N-1}b_{N-2} \dots b_1b_0 \quad (82)$$

où N dénote le nombre de bits.

La valeur R d'un nombre entier est calculée comme suit :

$$R = -2^{N-1}b_{N-1} + \sum_{i=0}^{N-2} 2^i b_i \quad (83)$$

La valeur R est limitée dans l'intervalle comme la suivante :

$$-2^{N-1} \leq R \leq 2^{N-1} - 1 \quad (84)$$

Un nombre fixe général consiste en une partie entière et une partie fractionnaire. Un nombre fixe signé de N bits avec I bits entiers et $N - I$ bits fractionnels peut être représenté comme suit :

$$b = \underbrace{b_{N-1}b_{N-2} \dots b_{N-I}}_{I \text{ bits}} \bullet \underbrace{b_{N-I-1}b_{N-I-2} \dots b_0}_{(N-I) \text{ bits}} \quad (85)$$

La valeur R d'un nombre fixe est calculée comme suit :

$$R = -2^{I-1}b_{N-1} + \sum_{i=0}^{I-2} 2^i b_{N-I+i} + \sum_{j=1}^{N-I} 2^{-j} b_{N-I-j} \quad (86)$$

La valeur R est limitée dans l'intervalle comme la suivante :

$$-2^{l-1} \leq R \leq 2^{l-1} + 1 - 2^{-(N-l)} \quad (87)$$

Un nombre fixe normalisé est un nombre fixe qui ne contient qu'un seul bit entier ; et peut être représenté comme suit :

$$b = b_{N-1} \bullet b_{N-2} \dots b_1 b_0 \quad (88)$$

Le bit entier est à la fois le MSB et le bit de signe. La valeur R d'un nombre fixe normalisé est calculée comme suit :

$$R = -b_{N-1} + \sum_{i=0}^{N-2} 2^{-i} b_i \quad (89)$$

Par distributivité du calcul précédent (81), un autre calcul possible de la valeur R d'un nombre fixe normalisé est le suivant :

$$R = \frac{-2^{N-1} b_{N-1} + \sum_{i=0}^{N-2} 2^i b_i}{2^{N-1}} \quad (90)$$

La valeur R est limitée dans l'intervalle comme la suivante :

$$-1 \leq R \leq 1 - 2^{-(N-1)} \quad (91)$$

Un opérateur arithmétique binaire des nombres entiers est simple. D'une manière générale, un opérateur arithmétique binaire suit la procédure suivante :

- Redistribution des bits (*bit casting*). Il est important de redistribuer les bits pour que l'opération soit sur le bon positionnement des bits, notamment dans le cas où le nombre de bits entiers et le nombre de bits fractionnaires de deux opérandes est respectivement égal.
- Opération arithmétique binaire sur des nombres binaires.

- Gestion du débordement de l'opération s'il y a lieu.
- Redistribution des bits sur le résultat. Cette opération a pour but d'assurer que le format des nombres respecte celui désiré pour le résultat. Il est possible qu'il y ait un débordement à cause de la redistribution des bits.

Il existe plusieurs méthodes pour gérer le débordement. Certes, la précision arithmétique change d'une méthode à l'autre. Le lecteur peut consulter [83] pour plus de détails. La méthode par saturation est couramment utilisée et a été choisie dans ce travail de thèse. Cette dernière consiste à forcer une valeur bornée (l'extrême de l'intervalle) que le format donné des nombres peut atteindre.

Les nombres entiers et les nombres fixes normalisés partagent le même comportement concernant l'arithmétique binaire. En effet, les opérations arithmétiques binaires sur des nombres entiers et des nombres fixes suivent la même procédure. Toutefois, les nombres fixes normalisés présentent une meilleure précision arithmétique par rapport aux nombres entiers dans certaines applications, notamment de traitement des signaux numériques. C'est pour cette raison que les nombres fixes normalisés sont adoptés pour certains processeurs DSP comme le DSP56002 de Motorola [80].

V.4.2. Paquetage VHDL en virgule fixe

Bien que certaines propositions des paquetages VHDL en virgule fixe existent tels que la proposition de standard IEEE [15] ou le paquetage de Duclos [35], ces propositions ne sont pas encore approuvées ou vérifiées au moment de la réalisation de nos travaux de recherches. Nous avons donc développé un paquetage VHDL en virgule fixe. Le but ultime de ce travail n'est pas de rivaliser avec la proposition de standard IEEE. En effet, nous avons développé ce paquetage pour une utilisation interne afin de faciliter nos implémentations d'algorithmes en virgule fixe, spécifiquement pour les cibles FPGA de Xilinx. Les différences majeures de notre paquetage vis-à-vis des autres propositions sont résumées dans le tableau 20, ce paquetage en virgule fixe est nommé *lssi_std_logic_fixed*.

Tableau 20 : Comparaison du paquetage des arithmétiques à virgule fixe *lssi_std_logic_fixed* par rapport aux autres propositions.

	Paquetage IEEE	Paquetage de Duclos	Paquetage du LSSI
Librairie de base	<p><i>numeric_std</i></p> <p>Ce paquetage permet de réaliser les arithmétiques sur des <i>signed</i> ou <i>unsigned</i>.</p> <p>Exemple d'application du <i>numeric_std</i>:</p> <pre>library IEEE; use IEEE.STD_LOGIC_1164.ALL; use IEEE.NUMERIC_STD.ALL; --... variable a : signed(7 downto 0); variable b : signed(7 downto 0); variable s: signed(7 downto 0); --... s := a + b; -- addition de deux nombres signés</pre>	Idem paquetage IEEE	<p><i>std_logic_arith</i></p> <p>Ce paquetage permet de réaliser les arithmétiques sur des <i>std_logic_vector</i>.</p> <p>Exemple d'application. du <i>std_logic_arith</i>:</p> <pre>library IEEE; use IEEE.STD_LOGIC_1164.ALL; use IEEE.STD_LOGIC_SIGNED.ALL; --... variable a : std_logic_vector(7 downto 0); variable b : std_logic_vector(7 downto 0); variable s: std_logic_vector(7 downto 0); --... s := a + b; -- addition de deux nombres signés</pre>
Modification/ajout particulier¹⁷	Nécessaire pour l'indexage négatif.	Idem IEEE	Aucune modification ou ajout aux paquetages standards n'est requise.
Indexage	<p>Un tableau avec indice d'un nombre négatif pour présenter un nombre fixe. Modification est nécessaire concernant l'indexage.</p> <p>Les bits avec indices positifs sont bits entiers. Les bits avec indices négatifs sont bits fractionnaires. Le MSB est le bit de signe.</p> <p>La présentation <i>little endian</i>.</p> <p>Exemple d'un nombre fixe signé de 8 bits avec 2 bits d'entier :</p> <pre>variable a signed(1 downto -6);</pre> <p>Dans l'exemple ci-haut, les bits 1 à 0 sont les bits entiers où le bit 1 est le signe. Les bits -1 à -6 sont les bits fractionnaires.</p>	Idem paquetage IEEE	<p>Un tableau avec indice ordinaire d'un nombre naturel pour présenter un nombre fixe. Aucune modification n'est nécessaire pour l'indexage.</p> <p>Les bits plus significatifs sont les bits entiers et le MSB est le signe si le nombre est signé. Les autres bits sont les bits fractionnaires.</p> <p>Idem paquetage IEEE</p> <p>Exemple d'un nombre fixe signé de 8 bits avec 2 bits d'entier :</p> <pre>variable a : std_logic_vector(7 downto 0);</pre> <p>Dans l'exemple ci-haut, bits 7 et 6 sont les bits entiers ou le bit 7 est le signe. Les bits 5 à 0 sont les bits fractionnaires.</p>
Cible techno.	Général	Général	FPGA, notamment Xilinx
Surplus matériel	Pas d'info. disponible	Pas d'info.	Minime
Synthèse	Général	Général	Optimisation ciblée sur les technologies FPGA, notamment Xilinx et Altera
Intégration dans le flot commerciale	Pas d'info.	Pas d'info	Leonardo pour l'instant.

Le développement de ce paquetage est basé sur le paquetage *std_logic_arith* de Synopsys© qui permet d'effectuer des arithmétiques sur *std_logic_vector*. La présente version contient des opérateurs de surcharge arithmétiques (addition, soustraction, multiplication), de conversion, et

¹⁷ On entend par « Modification/ajout particulier » comme une modification ou ajout particulier requis aux paquetages standards

de logique. De plus, la présente version traite le débordement des opérateurs arithmétique par saturation seulement.

Contrairement à la proposition IEEE qui introduit l'indice négatif d'un tableau des bits d'un nombre fixe, nous considérons l'indice non négatif. Le fait que l'on n'utilise que l'index non négatif permet d'effectuer des opérations arithmétiques sans introduire des conventions supplémentaires et le traitement relatif à ces dernières. Un nombre fixe du paquetage *lssi_std_logic_fixed* est déclaré comme suit dans le modèle VHDL :

$$\text{variable } a : \text{std_logic_vector}(N + I - 1 \text{ downto } I) ;$$

où N et I dénote respectivement le nombre de bits du variable à virgule fixe a et le nombre de bits entiers.

Ce paquetage a été développé lors de la modélisation en VHDL d'une architecture pour l'exécution d'un algorithme spécifique.

V.4.3. Classe de données en virgule fixe MATLAB® : AC_M

Dans cette section, nous proposons une classe de données en virgule fixe utilisée dans l'environnement MATLAB®. L'idée consiste à améliorer la méthodologie d'implémentation telle que vue dans l'approche unifiée à la figure 55 de la section V.3.3. Cette classe, nommée AC_M, est entièrement intégrée en environnement MATLAB®. Le principe est présenté à la figure 56 et se découpe en deux parties : partie pré-caractérisée et partie caractérisée par l'utilisateur. La partie pré-caractérisée consiste en fonctions déjà caractérisées et vérifiées. Elle sert plutôt au développement de la classe même. L'utilisateur de la classe s'intéresse plutôt à la partie caractérisée par l'utilisateur.

La partie caractérisée par l'utilisateur sert à définir la spécification des nombres. Les types de données ont été développés en se basant sur *templates*. Ces derniers obligent de passer les arguments comme constants puisque les arguments doivent être définis au temps de compilation par le compilateur. L'entête *ac_m_const.h* sert à spécifier les paramètres des nombres fixes. L'utilisateur n'a qu'à changer ces constantes dans cet entête. Le script MATLAB® *ac_m_compile.m*

sert à compiler les fonctions de passerelles Mex-C de la classe. L'exécution de ce script permet de passer les paramètres de nombres fixes selon l'intérêt de l'utilisateur aux fonctions de passerelles Mex-C de la classe AC_M.

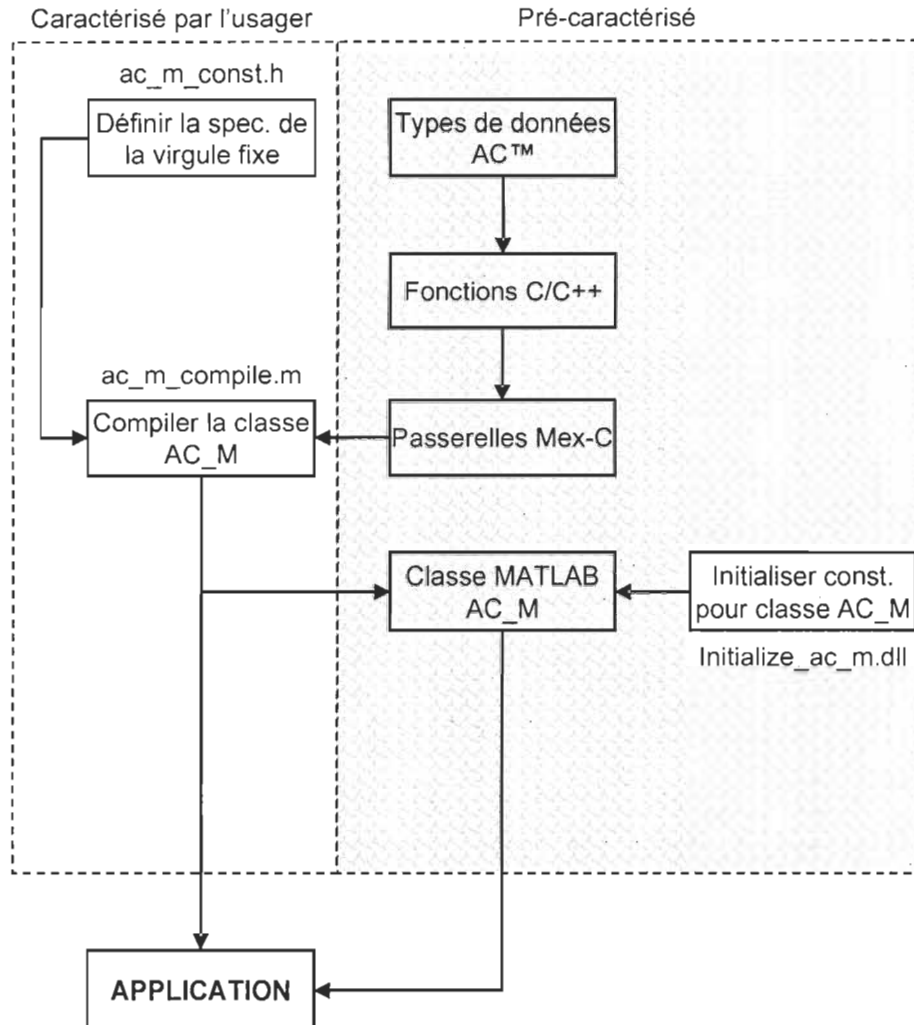


Figure 55 : Principe de la classe AC_M intégré sous MATLAB®.

La partie pré-caractérisée sert à définir la spécification des nombres fixes comme le nombre de bits, le nombre de bits entiers, le mode de quantification et le mode de débordement. Les composants de la partie pré-caractérisée sont les suivants :

- Les fonctions C/C++ ont été développées en C/C++ standard. Ces fonctions servent à réaliser les opérateurs arithmétiques des nombres fixes basés sur les types de données AC de Mentor Graphics™.
- Les passerelles Mex-C permettent d'exécuter les fonctions C/C++ en MATLAB®.
- Le DLL *Initialize_ac_m.dll* permet de passer les constantes définies par l'utilisateur dans l'entête *ac_m_const.h* en MATLAB® pour l'usage de la classe AC_M.

Le cœur de la classe AC_M consiste en des opérateurs arithmétiques de surcharge comme présentés au tableau 21. Une fois que les variables sont déclarées ou initialisées, on peut réaliser des opérateurs arithmétiques sur celles-ci.

Tableau 21 : Opérateurs arithmétiques de surcharge de la classe AC_M.

Opérateur	Syntaxe	Description
+	$a + b$	Addition
-	$a - b$ $- a$	Soustraction Soustraction unaire
*	$a * b$	Multiplication de deux matrices, ou multiplication d'un scalaire avec une matrice
.*	$a .* b$	Multiplication point par point de deux matrices, ou d'un scalaire avec une matrice
./	$a ./ b$	Division à droite point par point d'une matrice par une matrice, ou division d'une matrice par un scalaire
.\	$a .\ b$	Division à gauche point par point d'une matrice par une matrice ou d'une matrice par un scalaire.
.'	$a .'$	Transpose non conjugué
'	$a '$	Transpose conjugué
.^	$a .^n$	Puissance
sum	sum(a)	Sommation d'un vecteur ou d'une matrice. Si a est un vecteur, le résultat est un scalaire. Si a est une matrice, le résultat est un vecteur ligne dont chaque élément est la sommation de tous les éléments de la colonne correspondante.

En plus, il y a aussi les fonctions logiques comme dans le tableau 22, et les fonctions de conversion comme dans le tableau 23. Les opérateurs logiques et opérateurs de conversion ne sont disponibles que pour des données réelles. Les opérateurs arithmétiques sont fonctionnels sur les variables AC_M. Les fonctions logiques et de conversion sont fonctionnelles sur des variables ordinaires de MATLAB®.

Tableau 22 : Opérateurs logiques.

Opérateur	Syntaxe	Description
~	~a	Négation logique
&	a & b	Et logique
	a b	Ou logique

Tableau 23 : Opérateurs de conversion.

Opérateur	Syntaxe	Description
ac_m	ac_m(a)	Déclaration/conversion d'un variable AC_M dérivé du variable a.
get_data	get_data(a)	Prise de la partie de données d'un variable AC_M a. Cette commande est équivalente à l'opérateur double(a).
double	double(a)	Prise de la partie de données d'un variable AC_M a. Cette commande est équivalente à l'opérateur get_data(a).
quantize	quantize(a,WC,IC)	Quantification d'un nombre réel a en un nombre fixe avec nombre de bits WC et nombre de bits entier IC.
to_bin	to_bin(a,WC,IC)	Conversion d'un nombre réel en un nombre binaire avec nombre de bits WC et nombre de bits entier IC. Le nombre binaire est présenté sous forme d'une chaîne de caractères.
to_dec	to_dec(s,WC,IC)	Conversion d'un nombre binaire sous forme d'une chaîne de caractères en un nombre réel avec nombre de bits WC et nombre de bits entier IC.

Il est à noter que les fonctions C/C++ peuvent être directement appelées pour réaliser les opérateurs arithmétiques. Ces derniers sont présentés dans le tableau 23. Ces fonctions sont fonctionnelles en virgule fixe sur les variables ordinaires de MATLAB®. Les variables d'entrées peuvent être flottantes ou fixes quantifiés. Les résultats sont en nombres fixes quantifiés. Peu importe que les variables d'entrées soient quantifiés ou pas, elles seront quantifiées par les fonctions avant le calcul. La spécification des nombres fixes de ces fonctions est donnée dans l'entête *ac_m_const.h* (figure 56). Ces fonctions sont visuellement moins faciles à utiliser par rapport aux opérateurs de surcharge du tableau 21. Toutefois, elles sont beaucoup plus performantes en termes de vitesse de simulation comme nous le noterons dans les résultats d'évaluations qui suivent.

Tableau 24 : Fonctions C/C++ utilisables en environnement MATLAB®.

Fonctions C/C++ utilisables en environnement MATLAB®.		
Fonctions	Syntaxe	Description
ac_m_add	ac_m_add(a,b)	Addition
ac_m_sub	ac_m_sub(a,b)	Soustraction
ac_m_dotmult	ac_m_dotmult(a,b)	Multiplication point par point
ac_m_mult	ac_m_mult(a,b)	Multiplication matrice par matrice
ac_m_rdivide	ac_m_rdivide(a,b)	Division à droite
ac_m_sum	ac_m_sum(a)	Sommation

Dans le tableau 25, le benchmark consiste à exécuter deux cas étudiés : un égaliseur FIR avec adaptation LMS (FIR-LMS) [47] et le récepteur CF-MUD. Ces deux systèmes, FIR-LMS et CF-MUD, ont été modélisés en virgule fixe et simulés dans l'environnement MATLAB®. Ces deux cas d'études ont été modélisés en virgule fixe pour simuler dans l'environnement MATLAB®. Avec le filtre FIR-LMS, on a obtenu un gain 5 fois plus rapide en vitesse de simulation. Avec le récepteur CF-MUD, on a obtenu un gain de 2.

Tableau 25 : Benchmark avec un égaliseur FIR-LMS et un récepteur CF-MUD de la classe AC_M proposée par rapport à l'objet *fi* de MATLAB®.

Application	Gain de vitesse
Égaliseur avec filtrage FIR-LMS	5
Récepteur CF-MUD	2

Les tableaux 26 et 27 ont pour but de démontrer les performances de la classe AC_M vis-à-vis de l'objet MATLAB® *fi*. Dans ces travaux, nous avons considéré l'addition de deux vecteurs d'une longueur de 10000 éléments, la sommation d'un vecteur d'une longueur de 10000 éléments, la multiplication point-par-point de deux vecteurs d'une longueur de 10000 éléments et la multiplication matrice-par-matrice de deux matrices d'une dimension de 10000x10000 éléments. Les simulations ont été effectuées sur une machine portable Toshiba Tecra A3 avec le processeur Centrion à 1.6GHz et 1G RAM. Toutefois, ces résultats sont également valide sur une station une station IBM IntelliStation Z Pro avec un processeur Pentium® à 3.4 GHz et 2.75 GB RAM. Dans le tableau 26, le benchmark en exécution des opérations arithmétiques a été réalisé avec l'objet *fi* et la classe AC_M. On peut noter que la classe AC_M offre une amélioration en temps de simulation significative par rapport à l'objet *fi*.

Tableau 26 : Benchmark avec opérations arithmétiques de la classe AC_M proposée par rapport à l'objet *fi* de MATLAB®.

Opération arithmétique	Temps de simulation avec l'objet <i>fi</i> (s)	Temps de simulation avec la classe AC_M (s)	Gain de vitesse
Additions	0.5625	0.2500	3
Sommations	0.4688	0.0938	5
Multiplications point-par-point	1.4531	0.2500	6
Multiplications matrice-par-matrice	14.5156	0.9844	15

Dans le tableau 27, le benchmark consiste aussi à l'exécution des opérations arithmétiques. Cependant, par rapport aux résultats reportés au tableau 26, où les fonctions arithmétiques de la classe AC_M au tableau 21 sont utilisées, nous avons réalisé le tableau 27 à l'aide des fonctions

DLL de la classe AC_M au tableau 24. On peut constater qu'en utilisant les fonctions DLL, les performances en termes de temps de simulations augmentent davantage. Ceci peut être par le temps de transfert et de conversion de données dans la classe AC_M.

Tableau 27 : Benchmark avec opérations arithmétiques utilisant les DLLs de la classe AC_M proposée par rapport à l'objet *fi* de MATLAB®

Opération arithmétique avec DLLs	Temps de simulation avec l'objet <i>fi</i> (s)	Temps de simulation avec la classe AC_M (s)	Gain de vitesse
Additions	0.5938	0.0781	8
Sommations	0.3594	0.0781	5
Multiplications point par point	1.4531	0.0781	19
Multiplications matrice par matrice	14.4375	0.7656	19

V.5. Conclusions

Le temps de développement et de simulation des applications est significativement amélioré par les méthodologies de conception présentées dans ce chapitre. Nous avons proposé deux méthodologies de conception des systèmes complexes : l'approche conventionnelle et l'approche unifiée. Dans l'approche conventionnelle, le flot de conception est amélioré avec la capacité d'estimation des surfaces matérielles à partir d'une spécification architecturale. Dans l'approche unifiée, nous avons utilisé les langages de haut niveau – C/C++ et SystemC – dans le flot de conception. L'une est basée sur C/C++ et l'autre est basée sur SystemC respectivement. La méthode basée sur SystemC est possible par la proposition d'un cadre de cosimulation utilisant MATLAB® et SystemC. Ce cadre de cosimulation permet de simuler des modèles SystemC à l'intérieur de l'environnement de MATLAB®. Ce cadre de cosimulation a été développé avec une méthode d'interopérabilité entre l'environnement MATLAB® et l'environnement SystemC. Cette méthode a été validée par modélisation et cosimulation en virgule flottante et virgule fixe du récepteur Sign-MUD respectant les contraintes WCDMA – la Signature du récepteur

CF-MUD. L'autre méthode dans l'approche unifiée, consistant à utiliser le langage C++ standard dans le flot de conception, a été aussi présentée.

Pour supporter notre méthodologie de conception, nous avons développé un paquetage VHDL en virgule fixe nommé *lssi_std_logic_fixed* [RT2] et une classe de données en virgule fixe nommé AC_M MATLAB® [RT1]. Bien que d'autres propositions commerciales (l'objet *fi* de MATLAB par exemple) et de standards (la proposition IEEE pour VHDL par exemple) existent, nos développements s'avèrent nécessaires. En effet, l'utilisation des bibliothèques arithmétiques en virgule fixe déjà vérifiées, performantes et spécifiques pour les technologies cibles dans un flot de conception aide les concepteurs à économiser le temps consacré aux opérateurs arithmétiques. Pour le paquetage VHDL, comme les propositions existantes ne sont pas encore approuvées ou vérifiées, le paquetage *lssi_std_logic_fixed* a été alors proposé. De plus, ce paquetage est spécialisé pour les cibles FPGA Xilinx. Il offre certains avantages par le fait qu'il n'introduit pas de conventions supplémentaires et de traitement relatif sur l'indice des tableaux de bits. Pour la classe AC_M, elle a été développée à base des types de données Algorithmic C™ [70] qui accélère la vitesse de simulation par des *templates*. En utilisant la classe AC_M, la vitesse de simulation des noyaux d'algorithmes est beaucoup plus rapide par rapport à celle utilisant l'objet *fi* [67] de MATLAB®. Ces paquetages en virgules fixes – *lssi_std_logic_fixed* et AC_M – répondent à nos besoins spécifiques de développement au LSSI dans ce cadre de sa collaboration avec son partenaire industriel – la société de recherche Axiocom inc.

CHAPITRE VI

CONCLUSIONS ET PERSPECTIVES

En présence des interférences à accès multiple, la détection à usagers multiples devient un élément essentiel des systèmes de communications sans fil de la 3G. Elle permet d'annuler des interférences à la fois inter-symbole et à accès multiple. Les travaux algorithmiques de la MUD sont bien établis et documentés dans la littérature. Cependant, les travaux d'implémentation restent encore un sujet non-trivial. L'objectif principal de ce projet doctoral consistait donc à proposer et à développer les architectures parallèles d'une classe d'algorithmes adaptatifs de la détection à usagers multiples, nommée CF-MUD, développée au Laboratoire LSSI en collaboration avec son partenaire industriel – Axiocom inc.

Ainsi, les contributions de ce travail doctoral consistaient à :

- Proposer et développer les architectures VLSI d'une classe d'algorithmes d'annulation, nommée CF-MUD, ciblées sur FPGA.
- Étendre et implémenter l'algorithme CF-MUD existant pour les systèmes multi-débit de données.
- Proposer une méthodologie de conception clairement établie des systèmes complexes en utilisant les outils commerciaux. Ainsi, deux approches ont été élaborées : approche conventionnelle et approche unifiée. L'approche conventionnelle porte sur une amélioration dans le flot de conception standard. L'approche unifiée consiste à utiliser

les langages de haut niveau – C/C++, SystemC, MATLAB® – en conception matérielle des systèmes complexes.

VI.1. Contribution d'implémentation d'algorithmes spécifiques

Se basant sur un algorithme MUD favorable à l'implémentation, les architectures proposées sont différentes à d'autres propositions d'implémentation VLSI dans la littérature par ses performances et la faisabilité sur la cible des FPGA. Les architectures développées sont capables d'exécuter en temps réel plusieurs débits dans plusieurs scénarios de communication. Elles exploitent bien les technologies programmables modernes tels les FPGA. En plus, la différence avec d'autres travaux concerne le respect des contraintes de temps imposées par la norme de la 3G, plus spécifiquement celle du WCDMA, tout en considérant des limitations de la technologie ciblée. Bien que les architectures matérielles développées soient spécifiques, elles sont applicables aux autres cibles technologiques, voire ASIC, grâce à sa régularité.

Pour le récepteur CF-MUD mono-débit, on a démontré un taux d'utilisation des ressources près de 100% (ressources des multiplieurs câblés et blocs RAM) pour maximiser le parallélisme de l'algorithme. En plus, les travaux d'implémentation visent à maximiser le nombre d'utilisateurs sur un seul composant FPGA. Le nombre important d'utilisateurs intégrés sur un seul composant permet d'établir le nombre de composants FPGA nécessaire pour traiter plusieurs secteurs de la station de base de communication du réseau cellulaire. Aussi, on peut réduire considérablement le problème relatif aux communications entre composants FPGA. Le cœur de l'architecture a été modélisé avec environ 6000 lignes de codes VHDL synthétisables. Ces travaux nous ont permis de publier les articles [J1], [C3], [C4].

De plus, le récepteur CF-MUD a été étendu pour les systèmes hétérogènes de communication WCDMA et plus particulièrement dans un contexte multi-débit. Deux schémas ont été considérés : LRD et HRD. Bien que notre approche qui génère les utilisateurs virtuels soit différente de la littérature, les simulations ont démontré que nos résultats coïncident avec ceux des travaux existants [77]-[78]. En effet, les simulations numériques ont démontré que le schéma

LRD offre de meilleures performances par rapport au schéma HRD en termes de taux d'erreur binaire (BER). Toutefois, le schéma LRD introduit un surplus considérable des ressources vis-à-vis du schéma HRD. Nous avons analysé les performances matérielles de ces deux schémas multi-débit. Cette analyse est utile pour l'implémentation sur un processeur général, par exemple un processeur DSP. L'implémentation du récepteur MUD multi-débit a été réalisée en utilisant le noyau de récepteur CF-MUD mono-débit. Se basant sur l'architecture matérielle régulière du récepteur mono-débit, le développement et l'ajout en ressources de l'architecture du récepteur multi-débit étendu sont ainsi mineurs. Ces travaux nous ont permis de publier les articles [C1], [C2], ainsi que de soumettre l'article [J2].

VI.2. Contribution méthodologique

Les travaux d'implémentation des récepteurs à usagers multiples sophistiqués sont spécifiques à l'algorithmique dans la littérature. En effet, ils ne présentent pas de travail systématique du point de vue des méthodologies de conception. Cette ambiguïté nous a ainsi motivé aux travaux concernant les méthodologies de conception. Par conséquent, durant les travaux, nous avons développé deux méthodologies de conception clairement établies des systèmes complexes. L'une concerne une amélioration du flot conventionnel de conception [cf. V.2], et l'autre concerne une approche unifiée [cf. V.3].

Pour l'approche conventionnelle, utilisant plusieurs langages et plusieurs environnements de conception, nous avons apporté une amélioration dans le flot de conception. En effet, pendant le développement des spécifications architecturales d'un algorithme, on peut déjà estimer les ressources allouées. Ainsi, une méthode d'estimation des ressources matérielles FPGA basée sur un modèle de programmation non-linéaire en nombres entiers a été présentée [RT4] La méthode a été spécifiquement validée par une application des récepteurs pour les systèmes de communications WCDMA sur la cible Virtex-II and Virtex-II Pro. Toutefois, cette approche est applicable à un large éventail d'algorithmes réguliers et récurifs qui sont typiques dans le traitement numérique des signaux et des systèmes de communication. Cela est non seulement bénéfique pour des concepteurs matériels mais aussi pour des concepteurs algorithmiques. En

fait, l'estimation des ressources matérielles avant la modélisation offre aux concepteurs algorithmiques la possibilité d'estimer rapidement la complexité matérielle. Elle offre aux concepteurs matériels la possibilité d'estimer les ressources matérielles nécessaires d'une spécification architecturale avant la modélisation et la synthèse de ses modèles HDL. Par conséquent, elle contribue à diminuer le temps de développement. Dans ce travail, nous n'estimons que le nombre de Bloc RAM et de multiplieurs dédiés faute d'avoir accès à la base de données complète de ces cibles technologiques.

Pour l'approche unifiée, utilisant un seul langage et un seul environnement de conception, nous avons développé une méthodologie de conception utilisant les langages de haut niveau – C/C++, SystemC, MATLAB® – en conception des systèmes complexes [RT3]. Ainsi, une méthode de cosimulation SystemC/MATLAB® a été développée. Étant donné que nous ne voulons pas modifier le noyau du simulateur (ordonnanceur)¹⁸ de SystemC, la méthode d'appel développée permet de cosimuler des modèles en MATLAB® et en SystemC tout en demeurant dans l'environnement MATLAB®. L'autre direction dans notre approche unifiée consiste à utiliser le langage C/C++ purement standard en conception. Dans cette approche purement C/C++, nous n'avons pas besoin de simulateur particulier pour simuler des modèles C/C++ standard. L'environnement MATLAB® nous permet ce genre de simulation, avec peu de travail dans la passerelle (*gateway*). En effet, nous pouvons cosimuler des modèles en C/C++ et MATLAB® dans l'environnement MATLAB®. En plus, pour supporter notre méthodologie au niveau de la modélisation, nous avons développé les bibliothèques en virgule fixe en VHDL et MATLAB®. En effet, en utilisant ces bibliothèques déjà prédéveloppées, pré-vérifiées et spécifiques, nous pouvons accélérer davantage la modélisation des modèles des algorithmes d'intérêt.

VI.3. Application des résultats des travaux

Les résultats de ces travaux de recherche ont été directement mis à profit, notamment dans les travaux concernant le développement d'architectures matérielles des récepteurs avancés et le

¹⁸ *SystemC scheduler*(cf. [83])

développement d'une méthodologie systématique et rapide tenant compte de l'adéquation algorithme-architecture, au Laboratoire des signaux et systèmes intégrés à l'UQTR. Nous retrouvons en particulier:

- Les articles [J1] et [C3]-[C4] présentent les architectures VLSI des détecteurs à usagers multiples ciblées sur les familles FPGA Virtex-II et Virtex-II Pro de Xilinx®. Il s'agit de premier travail en littérature qui a pour contribution de maximiser le nombre d'usagers détectables sur un seul composant FPGA à base de l'algorithme CF-MUD. Ce travail a été réalisé en collaboration avec la compagnie Axiocom inc. spécialisée dans l'annulation des interférences pour les récepteurs sans fil de la 3G et plus. Ce projet a fait partie d'une entente cadre sur le transfert technologique en vue d'une commercialisation.
- Les articles [J2], [C1]-[C2] traitent de la détection à usagers multiples multi-débit qui permet de détecter les données de tous les usagers ayant plusieurs débits de données par un seul détecteur. Il s'agit de concevoir et d'implanter sur FPGA un détecteur basé sur le noyau CF-MUD.
- Étant donné la popularité de l'algorithme adaptatif LMS, spécifiquement aux applications des systèmes CDMA, le processeur traitant le filtrage adaptatif proposé dans le chapitre III est réutilisable dans l'implémentation matérielle d'un large éventail des algorithmes adaptatifs des systèmes CDMA.
- Les méthodologies développées dans le chapitre V contribuent aux flots de conception à la fois algorithmique et architecturale dans l'environnement MATLAB®.
- Les bibliothèques en virgule fixe en MATLAB [RT1] et en VHDL [RT2] sont applicables pour modéliser les modèles fonctionnels ainsi que les modèles matérielles des algorithmes de traitement des signaux.

Ces travaux réalisés dans le cadre d'une collaboration avec l'industrie (Axiocom inc.) ont servi à orienter les recherches vers des technologies brevetées répondant à des besoins spécifiques des communications cellulaires de la 3G et 4G.

VI.4. Perspectives de développements futurs

En ce qui concerne les perspectives de ce travail de thèse doctorale, les travaux développés durant la thèse ont apporté plusieurs ouvertures. En fait, l'insertion des architectures proposées aux infrastructures de BTS contribue à renforcer des performances de la reconstruction des données. De plus, l'expertise développée durant ces travaux de thèse est applicable à d'autres techniques avancées de reconstruction des données dans les systèmes de communications de futures générations tels que MIMO, *Beamforming*, etc. Cela fera une poursuite de mes travaux de recherche continue. En plus, l'approche unifiée de conception sera complétée par une synthèse des modèles de haut niveau. Aujourd'hui, les outils commerciaux sont déjà disponibles à ces fins. Nous connaissons alors des travaux de synthèse des modèles C/C++ des récepteurs radio, à l'aide de l'outil Catapult® [70], dans la littérature [116]. Ainsi, la synthèse des modèles synthétisables SystemC et C++ standard complétera l'approche unifiée.

PUBLICATIONS

Voici la liste partielle des articles et d'autres rédactions techniques publiés ou soumis dans le cadre des travaux de ma thèse.

Articles de journal publiés ou soumis

- [J1] Q.-T. Ho, D. Massicotte, and A.-O. Dahmane, "FPGA Implementation of an MUD Based on Cascade Filters for a WCDMA System", *EURASIP Journal on Applied Signal Processing*, vol. 2006, Article ID 52919, 12 pages, 2006.
- [J2] Q.-T. Ho, D. Massicotte, "Design and Implementation of an Adaptive Multiuser Detector for Multirate WCDMA Systems", soumis à *Journal on VLSI Signal Processing*, 2007.
- [J3] Q.-T. Ho, D. Massicotte, "A contemporary overview on hardware implementation of multiuser detection", en préparation pour un journal.

Articles de conférences publiés avec comité de lecture

- [C1] Q.-T. Ho, D. Massicotte, "Design and Implementation of an Adaptive Multiuser Detector for Multirate WCDMA Systems", IEEE International Midwest Symposium on Circuits and Systems 2007 (MWSCAS 2007), août 5-8 2007, Montréal, Canada, accepté.
- [C2] Q.-T. Ho, D. Massicotte, "Hardware Implementation Issues of Cascade Filters MUD for Multirate WCDMA Systems", IEEE Workshop on Signal Processing Systems 2005 (SiPS 2005), pp. 209-213, novembre 2-4, 2005, Athènes, Grèce.
- [C3] Q.-T. Ho, A.-O. Dahmane, D. Massicotte, "A Low Complexity Multiuser Detector and its FPGA Implementation for Wireless DS-CDMA Communication Systems", GSPx, septembre 27-30, 2004, Santa Clara, Californie, États-Unis.
- [C4] Q.-T. Ho, D. Massicotte, "FPGA Implementation of Adaptive Multiuser Detector for DS-CDMA Systems", International Conference on Field Programmable Logic and its applications 2004 (FPL'04), août 28-septembre 1, 2004, Anvers, Belgique.

Présentations et rapports techniques

- [PPT1] Q.-T. Ho, D. Massicotte, "Implantation des récepteurs à usagers multiples pour les communications sans fil de la 3G et 4G", 75^e Congrès de l'ACFAS, 9 mai 2007, Trois-Rivières, Québec.

- [PPT2] Q.-T. Ho, D. Massicotte, “Hardware Implementation Issues of Cascade Filters MUD for Multirate WCDMA Systems”, 2005 IEEE Workshop on Signal Processing Systems (SiPS 2005), novembre 2-4, 2005, Athènes, Grèce.
- [PPT3] Q.-T. Ho, D. Massicotte, “VLSI architecture of an adaptive multiuser detector for wireless DS-WCDMA communication and design methodologies of systems on a chip”, The Canadian Microelectronics Corporation (CMC) Microsystems 2005 annual symposium, Ottawa, octobre 13, 2005.
- [PPT4] Q.-T. Ho, D. Massicotte, “FPGA Implementation of Adaptive Multiuser Detector for DS-CDMA Systems”, International Conference on Field-Programmable Logic and its applications 2004 (FPL'04), août 31, 2004, Anvers, Belgique.
- [RT1] Q.-T. Ho, D. Massicotte, “Une classe de données en virgule fixe pour la simulation rapide des systèmes DSP complexes dans MATLAB”, Laboratoire LSSI, Département de génie électrique et génie informatique, Université du Québec à Trois-Rivières, décembre 2006, Trois-Rivières, Québec.
- [RT2] Q.-T. Ho, D. Massicotte, “Paquetage en virgule fixe synthétisable pour la conception des systèmes DSP STD_LOGIC_FIXED”, Laboratoire LSSI, Département de génie électrique et génie informatique, Université du Québec à Trois-Rivières, juin 2006, Trois-Rivières, Québec.
- [RT3] Q.-T. Ho, D. Massicotte “A Cosimulation Framework for WCDMA Receivers Using MATLAB and SystemC”, Laboratoire LSSI, Département de génie électrique et génie informatique, Université du Québec à Trois-Rivières, juin 2006, Trois-Rivières, Québec.
- [RT4] Q.-T. Ho, D. Massicotte, “FPGA Hardware Resource Estimation Based on an Integer Nonlinear Programming Model”, Laboratoire LSSI, Département de génie électrique et génie informatique, Université du Québec à Trois-Rivières, juin 2006, Trois-Rivières, Québec.

BIBLIOGRAPHIE

- [1] *3GPP TS 25.101 V7.2.0 User Equipment (UE) radio transmission and reception (FDD) (Release 7)*. 2007-05, Technical Specification Group Radio Access Network.
- [2] *3GPP TS 25.213 V7.2.0 Spreading and modulation (FDD) (Release 7)*. 2007-05, Technical Specification Group Radio Access Network.
- [3] *3GPP TS 25.211 V7.2.0 Physical channels and mapping of transport channels onto physical channels (FDD) (Release 7)*. 2007-05, Technical Specification Group Radio Access Network.
- [4] Adachi, F., M. Sawahashi, and H. Suda, *Wideband DS-CDMA for next-generation mobile communications systems*. Communications Magazine, IEEE, 1998. **36**(9): p. 56-69.
- [5] Airiau, R., et al. *VHDL, langage, modélisation, synthèse*. 2 ed. 1998: Presses Polytechniques et Universitaires Romandes.
- [6] Alam, F., *Simulation of Third Generation CDMA Systems*, MS thesis in *Electrical engineering*. 1999, Virginia Polytechnic Institute & State University: Blacksburg, Virginia.
- [7] Andrews, J.G., *Interference cancellation for cellular systems: a contemporary overview*. Wireless Communications, IEEE [see also IEEE Personal Communications], 2005. **12**(2): p. 19-29.
- [8] Baier, A., et al., *Design study for a CDMA-based third-generation mobile radio system*. Selected Areas in Communications, IEEE Journal on, 1994. **12**(4): p. 733-743.
- [9] Bhasker, J., *A SystemC Primer*. 2002: Star Galaxy Publishing.
- [10] Bhatt, T.M. and A.D. McCain, *Matlab as a development environment for FPGA design in Proceedings of the 42nd annual conference on Design automation 2005*, ACM Press: San Diego, California, USA.
- [11] Bening, L. and H.D. Foster, *Principles of Verifiable RTL Design*. 2001: Springer.
- [12] Berthelot, F., F. Nouvel, and D. Houzet. *Design methodology for runtime reconfigurable FPGA: From high level specification down to implementation*. in *SiPS 2005*. 2005.

- [13] _____, *Partial and dynamic reconfiguration of FPGA: a top down design methodology for an automatic implementation*. 2006.
- [14] Bickerstaff, M., et al. *A 24Mb/s radix-4 logMAP turbo decoder for 3GPP-HSDPA mobile wireless*. in *2003 IEEE International on Solid-State Circuits Conference (ISSCC 2003)*. 2003.
- [15] Bishop, D., *Create package(s) for fixed point math*. 2003, disponible sur www.vhdl.org
- [16] Bottomley, G.E., T. Ottosson, and Y.P.E. Wang, *A generalized RAKE receiver for interference suppression*. *Selected Areas in Communications, IEEE Journal on*, 2000. **18**(8): p. 1536-1545.
- [17] Boutillon, E., W.J. Gross, and P.G. Gulak, *VLSI architectures for the MAP algorithm*. *Communications, IEEE Transactions on*, 2003. **51**(2): p. 175-185.
- [18] Bouchhima F. et al., *Discrete-continuous simulation model for accurate validation in component-based heterogeneous SoC design*. in *IEEE International Workshop on Rapid System Prototyping 2005 (RSP05)*, Montreal, Canada, 2005, pp. 181-187.
- [19] Brennan, D.G., *Linear diversity combining techniques*. IRE, 1959. **47**: p. 1075-1102.
- [20] Buehrer, R.M., N.S. Correal-Mendoza, and B.D. Woerner, *A simulation comparison of multiuser receivers for cellular CDMA*. *Vehicular Technology, IEEE Transactions on*, 2000. **49**(4): p. 1065-1085.
- [21] Chang, Y.-j., Y.-N. Lin, and D.W. Lin, *DSP implementation of successive interference cancellation (SIC) receiver for 3GPP WCDMA uplink transmission*. *Wireless Communications and Mobile Computing*, 2003. **3**(6): p. 789–800.
- [22] Chen, C., et al. *Rapid design and analysis of communication systems using the BEE hardware emulation environment*. 2003.
- [23] Chen, J. and U. Mitra. *Further results for multi-rate decorrelators for synchronous DS/CDMA systems*. in *34th Annual Allerton Conference on Communication, control and computing*. 1996. Monticello, IL.
- [24] Cheol, M., C. Myung-Sun, and P. Han-Kyu, *Performance of 2-D RAKE receiver in a correlated frequency-selective Nakagami-fading*. *Vehicular Technology, IEEE Transactions on*, 2001. **50**(5): p. 1312-1317.
- [25] Chih-Lin, I. and R.D. Gitlin. *Multi-code CDMA wireless personal communications networks*. 1995.
- [26] Ching-Hung, C., et al., *A programmable pipelined digital differential matched filter for DSSS receiver*. *Selected Areas in Communications, IEEE Journal on*, 2001. **19**(11): p.

- 2142-2150.
- [27] Chryssomallis, M., *Smart Antennas*, in *IEEE Antennas and Propagation Magazine*. 2000. p. 129-136.
 - [28] Correal, N.S., R.M. Buehrer, and B.D. Woerner, *A DSP-based DS-CDMA multiuser receiver employing partial parallel interference cancellation*. Selected Areas in Communications, IEEE Journal on, 1999. **17**(4): p. 613-630.
 - [29] Constantinides, G.A., P.Y.K. Cheung, and W. Luk, *Wordlength optimization for linear digital signal processing*. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 2003. **22**(10): p. 1432-1442.
 - [30] Dahmane, A.O., *Méthode de détection à usagers multiples pour les systèmes de communication DS-CDMA*, Thèse doctoral en génie électrique. 2004, Université du Québec à Trois-Rivières: Trois-Rivières.
 - [31] Das, S., et al., *Arithmetic Acceleration Techniques for Wireless Communication Receivers*, in *33rd Asilomar Conference on Signals, Systems and Computers*. 1999: Pacific Grove, CA.
 - [32] Dinan, E.H. and B. Jabbari, *Spreading codes for direct sequence CDMA and wideband CDMA cellular networks*. Communications Magazine, IEEE, 1998. **36**(9): p. 48-54.
 - [33] Divsalar, D., M.K. Simon, and D. Raphaeli, *Improved parallel interference cancellation for CDMA*. Communications, IEEE Transactions on, 1998. **46**(2): p. 258-268.
 - [34] Dubois M., Aboulhamid E. M, *Techniques to improve cosimulation and interoperability of heterogeneous models*, the *12th IEEE International Conference on Electronics, Circuits and Systems (ICECS2005)*. Gammarth, Tunisia, 2005.
 - [35] Duclos, *Synthesisable fixed-point package fix_std*, dernière consultation en 09é2006, disponible sur <http://www.duclos.com>.
 - [36] Eltawil, A.M. and B. Daneshrad, *A low-power DS-CDMA RAKE receiver utilizing resource allocation techniques*. Solid-State Circuits, IEEE Journal of, 2004. **39**(8): p. 1321-1330.
 - [37] Esmailzadeh, R. and M. Nakagawa, *TDD-CDMA for Wireless Communications*. 2002: Artech House Publishers.
 - [38] Fanucci, L., et al., *Design, implementation and verification through a real-time test-bed of a multi-rate CDMA adaptive interference mitigation receiver for satellite communication*. International Journal of Satellite Communications and Networking, 2003. **21**: p. 39-64.

- [39] _____, *VLSI implementation of a CDMA blind adaptive interference-mitigating detector*. Selected Areas in Communications, IEEE Journal on, 2001. **19**(2): p. 179-190.
- [40] Fisher, J.A., P. Faraboschi, and C. Young, *Embedded Computing: A VLIW Approach to Architecture, Compilers and Tools*. 2005: Morgan Kaufmann.
- [41] G.A.U.T, disponible sur : <http://web.univ-ubs.fr/lester/www-gaut/>, dernière consultation en 07/2007.
- [42] Gebotys, C.H. and M.I. Elmasry, *Optimal VLSI architectural synthesis: area, performance and testability*. 1992: Kluwer Academic Publishers.
- [43] Gershman, A.B. and N.D. Sidiropoulos, *Space-Time Processing for MIMO Communications*. 2005: John Wiley & Sons Ltd.
- [44] Golomb, S.W., *Shift Register Sequences*. Revised ed. 1982: Aegean Park Press.
- [45] Guoqiang, X., et al., *Adaptive multistage parallel interference cancellation for CDMA*. Selected Areas in Communications, IEEE Journal on, 1999. **17**(10): p. 1815-1827.
- [46] Han, K., *Automating Transformations from Floating-point to Fixed-point for Implementing Digital Signal Processing Algorithm*, Ph.D. dissertation in *Electrical Engineering*. 2006, University of Texas at Austin: Austin.
- [47] Haykin, S., *Adaptive Filter Theory*. 4 ed. 2001: Prentice Hall.
- [48] Hennessy, J.L. and D.A. Patterson, *Computer Architecture: A Quantitative Approach*. 3 ed. 2002: Morgan Kaufmann.
- [49] Ho, Q.-T., D. Massicotte, and A.-O. Dahmane, *FPGA Implementation of an MUD Based on Cascade Filters for a WCDMA System*. EURASIP Journal on Applied Signal Processing, 2006. **2006**: p. Article ID 52919, 12 pages.
- [50] _____, *A Low Complexity Adaptive Multiuser Detector and FPGA Implementation for Wireless DS-WCDMA Communication Systems*. in *GSPx 2004*. 2004. Santa Clara, CA, U.S.A.
- [51] Ho, Q.-T. and D. Massicotte. *FPGA Implementation of Adaptive Multiuser Detector for DS-CDMA Systems*. in *14th International Conference on Field-Programmable Logic and its applications (FPL 2004)*. 2004. Antwerp, Belgium.
- [52] _____. *Hardware implementation issues of cascade filters MUD for multirate WCDMA systems*. in *IEEE Workshop on Signal Processing Systems Design and Implementation (SiPS05)*. 2005. Athens, Greece. p. 204-208.
- [53] _____, *Design and Implementation of an Adaptive Multiuser Detector for Multirate*

- WCDMA Systems*, IEEE International Midwest Symposium on Circuits and Systems 2007 (MWSCAS 2007), août 5-8 2007, Montréal, Canada, accepté.
- [54] Texas Instruments, *Implementation of a WCDMA Rake Receiver on a TMS320C62xE DSP Device*, application report. 2000.
 - [55] Jiangxin, C. and U. Mitra, *Analysis of decorrelator-based receivers for multirate DS/CDMA communications*. Vehicular Technology, IEEE Transactions on, 1999. **48**(6): p. 1966-1983.
 - [56] Jinho, C. and K. Seong Rag, *Adaptive MMSE receiver for multirate CDMA systems*. Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on], 2002. **50**(12): p. 3098-3106.
 - [57] Jones, B.A. and J.R. Cavallaro, *A Rapid Prototyping Environment for Wireless Communication Embedded Systems*. EURASIP Journal on Applied Signal Processing, 2003. **2003**(6): p. 603-614.
 - [58] Karim, M.R. and M. Sarraf, *W-CDMA and cdma2000 for 3G Mobile Networks*. 2002: McGraw-Hill Professional.
 - [59] Keating, M. and P. Bricaud, *Reuse Methodology Manual for SoC Designs*. 3 ed. 2002: Kluwer Academic Publishers.
 - [60] Ki-II, K., K. Jiyang, and S. Wonyong, *AUTOSCALER for C: an optimizing floating-point to integer C program converter for fixed-point digital signal processors*. Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on], 2000. **47**(9): p. 840-848.
 - [61] Lee, B.G. and B.-H. Kim, *Scrambling Techniques for CDMA Communications*. International Series in Engineering & Computer Science. 2001: Kluwer Academic Publishers.
 - [62] Lequepeys, J.R., et al. *CESSIUM: a single component for implementing high data rates DSSS/CDMA interference cancellation receivers*. in *IEEE 5th International Symposium on Spread Spectrum Techniques and Applications*. 1998.
 - [63] Leung, O., T. Chi-Ying, and R.S. Cheng. *VLSI implementation of rake receiver for IS-95 CDMA testbed using FPGA*. in *Asia and South Pacific Design Automation Conference 2000 (ASP-DAC 2000)*. 2000.
 - [64] Lippman, S.B. and J. Lajoie, *C++ Primer*. 3 ed. 1998: Addison-Wesley Professional.
 - [65] Litva, J. and T.K.-Y. Lo, *Digital Beamforming in Wireless Communications*. 1996: Artech House Publishers.

- [66] Massicotte, D., *Une approche à l'implantation en technologie VLSI d'une classe d'algorithmes de reconstitution de signaux*, Thèse doctorat en génie électrique. 1995, École Polytechnique de Montréal.
- [67] Mathworks, *Fixed-Point Toolbox 2.0*, dernière consultation en 09/2006, disponible sur www.mathworks.com.
- [68] _____, *Link for ModelSim®*, dernière consultation en 09/2006, disponible sur www.mathworks.com.
- [69] _____, *External Interfaces*, dernière consultation en 09/2006, disponible sur : www.mathworks.com.
- [70] Mentor Graphics, *Algorithmic C™ Datatypes*, dernière consultation en 09/2006, disponible sur www.mentor.com.
- [71] _____, *Catapult® Synthesis*, dernière consultation en 09/2006, disponible sur www.mentor.com.
- [72] Menard, D., et al. *Efficient implementation of a rake receiver on the TMS320C64x*. in *Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*. 2003.
- [73] Micheli, G.D., *Synthesis and Optimization of Digital Circuits*. 1994: McGraw-Hill.
- [74] Ming, C. *Parallel systolic array implementation of multiuser detection for asynchronous DS/CDMA*. 2001.
- [75] Mingshu, W., L. Jianhua, and Z. Zucheng. *A novel interference cancellation receiver in VSL-CDMA systems*. 2003.
- [76] Mingxi, F., C. Hoffmann, and S. Kai-Yeung, *Error-rate analysis for multirate DS-CDMA transmission schemes*. Communications, IEEE Transactions on, 2003. **51**(11): p. 1897-1909.
- [77] Mitra, U., *Comparison of maximum-likelihood-based detection for two multirate access schemes for CDMA signals*. Communications, IEEE Transactions on, 1999. **47**(1): p. 64-77.
- [78] _____. *Further results for multi-rate decorrelators for synchronous DS/CDMA systems*. in *Proceedings of 34th Annual Allerton Conference on Communication, control and computing*. 1996. Monticello, IL.
- [79] Moshavi, S., *Multi-user detection for DS-CDMA communications*. Communications Magazine, IEEE, 1996. **34**(10): p. 124-136.
- [80] Motorola, *DSP 56002 data sheet*, dernière consultation en 09/2006, disponible sur :

www.freescale.com/

- [81] Nash, S.G. and A. Sofer, *Linear and Nonlinear Programming*. 1996: McGraw-Hill.
- [82] Ojanperä, T. and R. Prasad, *An Overview of Air Interface Multiple Access for IMT-2000/UMTS*, in *IEEE Communications Magazine*. 1998. p. 82-95.
- [83] OSCI, *IEEE Standard SystemC® Language Reference Manual*, SystemC official website, available at: <http://www.systemc.org>, last access Mar 2006.
- [84] Pahalawatha, N.C., et al., *Real time implementation of a MIMO adaptive power system stabiliser*. Generation, Transmission and Distribution [see also IEE Proceedings-Generation, Transmission and Distribution], IEE Proceedings C, 1990. **137**(3): p. 186-194.
- [85] Pedersen, K.I., et al. *Practical implementation of successive interference cancellation in DS/CDMA systems*. 1996.
- [86] picoChip, *picoArray™*, dernière consultation en 06/2007, disponible sur: www.picochip.com
- [87] Prasad, R. and T. Ojanpera. *A survey on CDMA: evolution towards wideband CDMA*. in *IEEE 5th International Symposium on Spread Spectrum Techniques and Applications*. 1998.
- [88] Prosilog, *Prosilog SystemC Compiler Ver 1.0*. 2002: , dernière consultation en 09/2006, disponible sur : www.prosilog.com.
- [89] Rajagopal, S., *Data-Parallel Digital Signal Processors: Algorithm Mapping, Architecture Scaling and Workload Adaptation*, Ph.D. dissertation in *Electrical and computer engineering*. 2004, Rice: Houston.
- [90] Rajagopal, S., et al., *Real-time algorithms and architectures for multiuser channel estimation and detection in wireless base-station receivers*. Wireless Communications, IEEE Transactions on, 2002. **1**(3): p. 468-479.
- [91] _____, *Efficient VLSI Architectures for Multiuser Channel Estimation in Wireless Base-Station Receivers*. Journal of VLSI Signal Processing, 2002. **31**: p. 143–156.
- [92] Rissa, T., A. Donlin, and W. Luk. *Evaluation of SystemC modelling of reconfigurable embedded systems*. in *2005 Design, Automation and Test in Europe (DATE'05)*. 2005. MESSE Munich, Germany.
- [93] Sabharwal, A., U. Mitra, and R. Moses, *MMSE receivers for multirate DS-CDMA systems*. Communications, IEEE Transactions on, 2001. **49**(12): p. 2184-2197.

- [94] Saquib, M., R.D. Yates, and A. Ganti, *An asynchronous multirate decorrelator*. Communications, IEEE Transactions on, 2000. **48**(5): p. 739-742.
- [95] Sawahashi, M., et al., *Experiments on pilot symbol-assisted coherent multistage interference canceller for DS-CDMA mobile radio*. Selected Areas in Communications, IEEE Journal on, 2002. **20**(2): p. 433-449.
- [96] Seepold, R. and A. Kunzmann, *Reuse Techniques for VLSI Design* 1ed. 1999: Springer.
- [97] Sengupta, C., *Algorithms and Architectures for Channel Estimation in Wireless CDMA Communication Systems*, Ph.D. dissertation in *Electrical and Computer Engineering*. 1998, Rice University: Houston.
- [98] Seskar, I.P. and N.B. Mandayam, *A software radio architecture for linear multiuser detection*. Selected Areas in Communications, IEEE Journal on, 1999. **17**(5): p. 814.
- [99] Sgroi, M., *Platform-based Design methodologies for Communication Networks*, Ph.D. dissertation in *Electrical Engineering and Computer Sciences*. 2002, University of California, Berkeley.
- [100] She-Hwa, Y. and W. Chorng-Kuang. *A 2 V CMOS programmable pipelined digital differential matched filter for DS-CDMA system*. 1999.
- [101] Siemens AG, *TD-SCDMA: the Solution for TDD bands*. 2001, White-paper.
- [102] Synopsys, *CoCentric® SystemC™ Compiler: RTL User and Modeling Guide*. Vol. Version U-2003.06, June 2003. 2003: Synopsys.
- [103] Tachikawa, E.K., *W-CDMA*. 2002: John Wiley & Sons, Ltd.
- [104] Taylor, F. and J. Mellot, *Hands-on digital signal processing*. 1998: McGraw-Hill.
- [105] Umamageswaran, K., S.L. Pandey, and P.A. Wilsey, *Formal Semantics and Proof Techniques for Optimizing VHDL Models*. 1999: Springer.
- [106] Van Meeteren, A.F., et al. *Groupwise weighted parallel interference cancellation for asynchronous multirate DS-CDMA*. 1999.
- [107] Varanasi, M.K. and B. Aazhang, *Multistage detection in asynchronous code-division multiple-access communications*. Communications, IEEE Transactions on, 1990. **38**(4): p. 509-519.
- [108] Vembu, S. and A.J. Viterbi. *Two different philosophies in CDMA-a comparison*. 1996.
- [109] Verdú, S., *Multiuser detection*. 1998, New York: Cambridge University Press.
- [110] Viterbi, A.J., *Cdma: Principles of Spread Spectrum Communication*. 1995: Addison-

Wesley.

- [111] Wijting, C.S., et al. *Groupwise serial multiuser detectors for multirate DS-CDMA*. 1999.
- [112] Xilinx, *Xilinx System Generator™ for DSP*, dernière consultation en 07/2007, disponible sur www.xilinx.com.
- [113] _____, *AccelDSP™*, dernière consultation en 07/2007, disponible sur www.xilinx.com.
- [114] Xu, G., et al., *VLSI Implementation of the Multistage Detector for Next Generation Wideband CDMA Receivers*. Journal of VLSI Signal Processing, 2002. **30**: p. 21-33.
- [115] Yingwei, Y., H.V. Poor, and S. Feng-Wen, *User capacity for synchronous multirate CDMA systems with linear MMSE receivers*. Information Theory, IEEE Transactions on, 2004. **50**(11): p. 2785-2793.
- [116] Yuanbin, G. and R.C. Joseph, *A Low Complexity and Low Power SoC Design Architecture for Adaptive MAI Suppression in CDMA Systems*. The Journal of VLSI Signal Processing, 2006. **V44**(3): p. 195-217.
- [117] Zeng, M., A. Annamalai, and V.K. Bhargava, *Recent advances in cellular wireless communications*. Communications Magazine, IEEE, 1999. **37**(9): p. 128-138.
- [118] Zhang, N., et al. *Trade-offs of performance and single chip implementation of indoor wireless multi-access receivers*. 1999.
- [119] Zihua, G. and K.B. Letaief, *Performance of multiuser detection in multirate DS-CDMA systems*. Communications, IEEE Transactions on, 2003. **51**(12): p. 1979-1983.

ANNEXE A

CARACTÉRISTIQUES RADIO ESSENTIELLES DES SYSTÈMES WCDMA MODE FDD

Comme les travaux de recherche dans ce manuscrit visent aux BTS des systèmes WCDMA, cette annexe sert à synthétiser les caractéristiques radio essentielles dans la couche physique (*layer 1*) de la liaison montante (*uplink*) de la norme WCDMA. La norme WCDMA supporte les deux modes de duplex, soit TDD et FDD. Dans tous les deux modes, les données sont transmises en trames de 10ms, et la bande passante est de 5MHz. Dans le mode FDD, deux bandes fréquentielles symétriques différentes sont utilisées pour la liaison montante et descendante. Dans le mode TDD, la même bande fréquentielle est utilisée et répartie pour les deux sens de transmission (une moitié pour liaison montante et une moitié pour la liaison descendante) [58], [117]. On s'intéresse ici au mode FDD.

Les paramètres de conception les plus importants du WCDMA sont résumés dans le tableau 28. Ainsi, la norme WCDMA se base sur la modulation DS-CDMA avec la largeur de la bande passante de 5 MHz. La fréquence des brides est à 3.84 MHz. Le débit de données est déterminé par le gain OVSF. La fonction de détection à usagers multiples est recommandée par la norme mais optionnelle dans l'implémentation pratique des systèmes. La WCDMA supporte le transfert intercellulaire doux (*soft handover*). La dernière est possible car les cellulaires utilisent la même fréquence de porteuse. Le contrôle de puissance avec boucle fermée (mesure-commande-réaction) à la fréquence de 1500 par seconde (1,5 KHz).

Tableau 28 : Paramètres de conception importants de la norme WCDMA.

Méthode d'accès	DS-CDMA
Largeur de bande	5MHz
Débit des brides	3,84 Mbrides/s
Méthode de duplex	FDD ou TDD
Longueur de trame	10ms
Conception des brides à multi-débit	Gain d'étalement variable
<i>Soft handover</i>	Oui
Contrôle de puissance	1,5 KHz pour liaison montante ou liaison descendante
Détection	Cohérent pour liaison montante ou liaison descendante
Détection à usagers multiples, antenne intelligente	Supportée par le standard, optionnelle lors de l'implémentation
Diversité	5MHz donne une diversité multi-trajet avec un récepteur <i>Rake</i>

A.1. Structure du canal physique

La structure d'une trame (*frame*) est présentée à la figure 57 [3]. Une trame de données se compose d'un canal de données (*DPDCH : Dedicated physical data channel*) et d'un canal de contrôle (*DPCCH : Dedicated physical control channel*). Chaque trame est d'une longueur de 38400 brides, et divisée en 15 *slots* dans lesquels chacun est d'une longueur de 2560 brides.

Le canal DPDCH transporte les données dont le nombre de bits-symboles de chaque *slot* est déterminé par le facteur d'étalement : $N_{\text{data}} = 10(256/N_c)$ bits, où $N_c = 2^k$ ($k = 0, 1, \dots, 6$) est le facteur d'étalement.

Le canal DPCCH transporte les données de contrôle. Chaque *slot* se compose de quatre champs : Pilot, TFCI (*Transport format combination indicator*), FBI (*Feedback information*) et TPC (*Transmit power control*). Il y a un seul canal DPCCH sur chaque liaison radio. Les champs du canal DPCCH sont définis comme ci-dessous.

- Le champ *Pilot* contient les bits connus qui sont transmis pour supporter l'estimation cohérente du canal.
- Le champ *TFCI* est optionnel. Il a pour but d'informer le récepteur sur la combinaison de format des canaux simultanés de transports mappés sur la trame *DPDCH*.
- Le champ *FBI* est utilisé pour supporter les techniques qui requièrent la rétroaction de la station utilisateur (*UE : User equipment*) au point d'accès UTRAN.
- Le champ *TPC* est utilisé pour le contrôle de puissance.

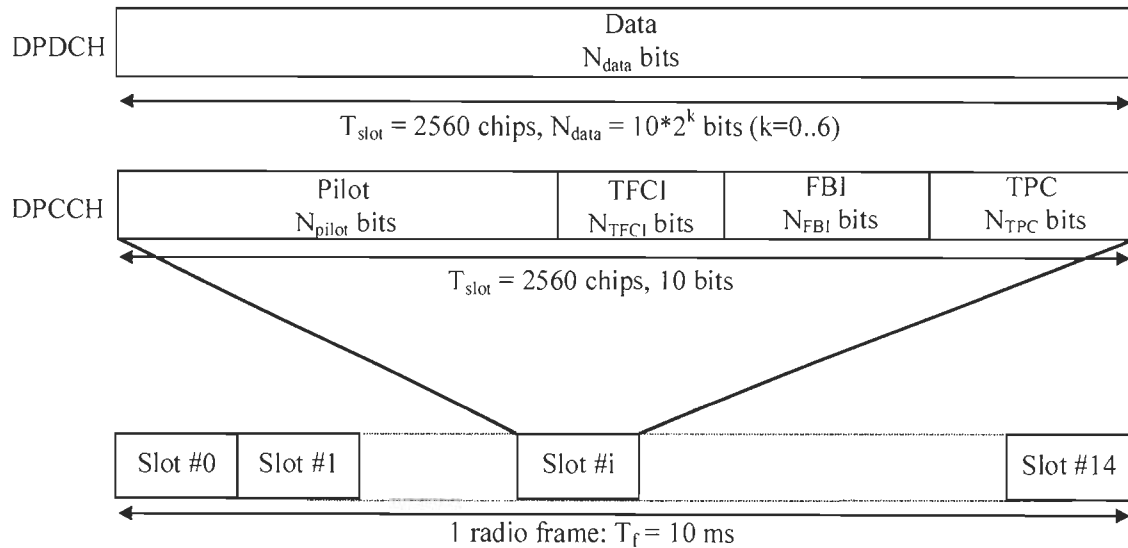


Figure 56 : Structure d'une trame de la liaison montante.

Le nombre de bits-symboles dans chaque *slot* d'une trame de données *DPDCH* (N_{data}) varie en fonction de débit de données qui est déterminé par le facteur d'étalement. Le nombre de bits dans chaque canal de contrôle *DPCCH* est toujours à 10 bits, parce que le facteur d'étalement est toujours à 256. Le nombre de bits-pilots (N_{pilot}) varie entre 3 et 8. Le nombre de

bits $TFCI$ (N_{TFCI}) varie entre 0 et 2. Le nombre de bits FBI (N_{FBI}) varie entre 0 et 2. Le nombre de bits TPC (N_{TPC}) varie entre 1 et 2.

A.2. Modulation et étalement de spectre

La figure 58 illustre le principe de l'étalement et de la modulation de la liaison montante [2], [6]. L'étalement est appliqué aux canaux physiques – données et contrôle – et présenté à la figure 58.a. Il consiste en deux opérations : l'opération de canalisation et l'opération de brouillage. La canalisation consiste à transformer chaque symbole de données en un nombre de brides. Il s'agit d'étaler le symbole avec un code-canalisation. Le nombre de brides par symbole est égal au facteur d'étalement. L'opération de brouillage consiste à multiplier un code de brouillage avec le signal étalé. Tandis que le code-canalisation est commun pour tous les usagers, le code de brouillage est distinct et unique pour chaque usager. Ainsi, le code-signature consiste en la multiplication de code-canalisation avec le code-brouillage. La modulation, présentée à la figure 58.b, consiste à moduler le signal de brouillage résultant avec le facteur *roll-off* de 0.22.

Les codes-canalisation sont les codes OVSF qui conservent l'orthogonalité entre usagers des canaux physiques différents. Les codes OVSF peuvent être définis utilisant un arbre de codes comme présenté à la figure 59. Dans cette figure, les codes de canalisation sont décrits comme $C_{ch,SF,k}$, où SF dénote le facteur d'étalement; et k , avec $0 \leq k \leq SF-1$, dénote le numéro de code. Chaque niveau de l'arbre des codes définit les codes de canalisation de longueur SF . La valeur à l'extrême gauche de chaque code correspond à la bride qui est premièrement transmis. Le nombre de codes pour un facteur SF particulier est égal à ce facteur d'étalement même. Tous les codes d'un même niveau constituent un ensemble et sont orthogonaux. Une paire quelconque de deux codes de niveaux différents sont orthogonaux à condition que l'un ne soit pas racine de l'autre.

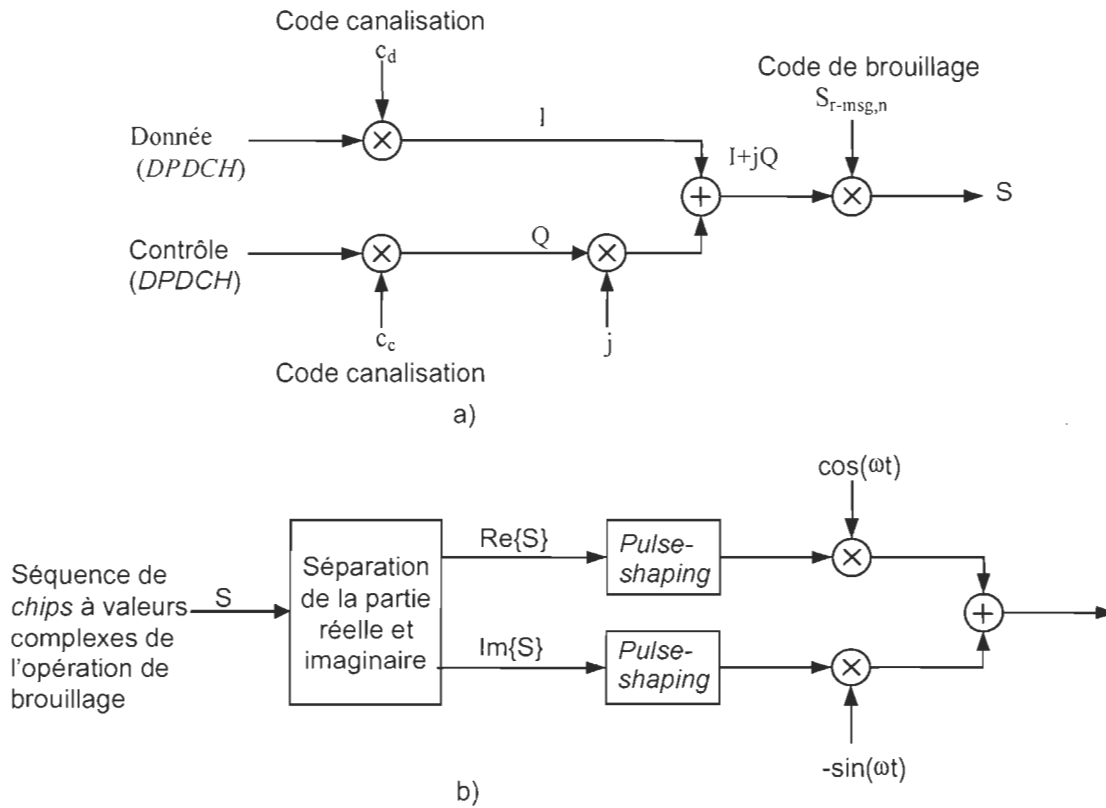


Figure 57 : Étatement et modulation de la liaison montante : a) Étatement, b) Modulation.

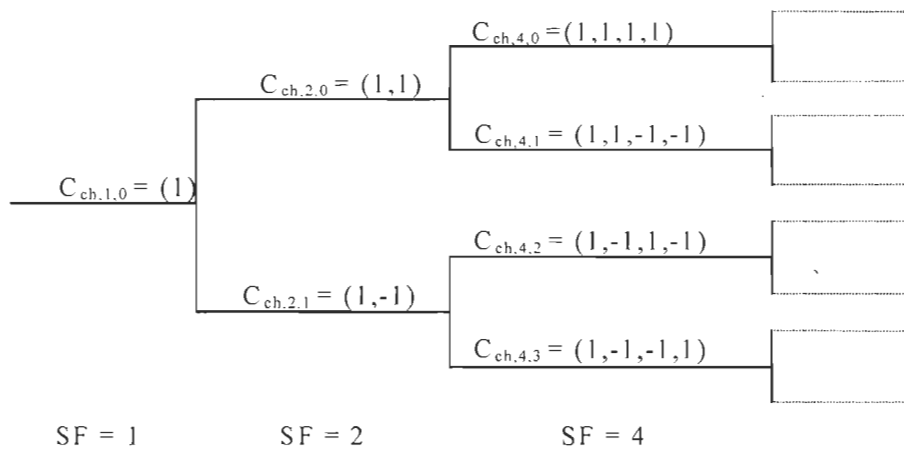


Figure 58: Arbre de codes OVSF.

La méthode de génération des codes de canalisation est définie comme suit :

$$\begin{aligned}
 C_{ch,1,0} &= 1, \\
 \begin{bmatrix} C_{ch,2,0} \\ C_{ch,2,1} \end{bmatrix} &= \begin{bmatrix} C_{ch,1,0} & C_{ch,1,0} \\ C_{ch,1,0} & -C_{ch,1,0} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \\
 \begin{bmatrix} C_{ch,2^{(n+1)},0} \\ C_{ch,2^{(n+1)},1} \\ C_{ch,2^{(n+1)},2} \\ C_{ch,2^{(n+1)},3} \\ \vdots \\ C_{ch,2^{(n+1)},2^{(n+1)}-2} \\ C_{ch,2^{(n+1)},2^{(n+1)}-1} \end{bmatrix} &= \begin{bmatrix} C_{ch,2^n,0} & C_{ch,2^n,0} \\ C_{ch,2^n,0} & -C_{ch,2^n,0} \\ C_{ch,2^n,1} & C_{ch,2^n,1} \\ C_{ch,2^n,1} & -C_{ch,2^n,1} \\ \vdots & \vdots \\ C_{ch,2^n,2^n-1} & C_{ch,2^n,2^n-1} \\ C_{ch,2^n,2^n-1} & -C_{ch,2^n,2^n-1} \end{bmatrix}
 \end{aligned}$$

Les codes de brouillage peuvent être générés par un registre à décalage. La figure 60 illustre une structure d'implémentation du générateur des séquences de codes courts de brouillage à l'aide de trois registres à décalage [2].

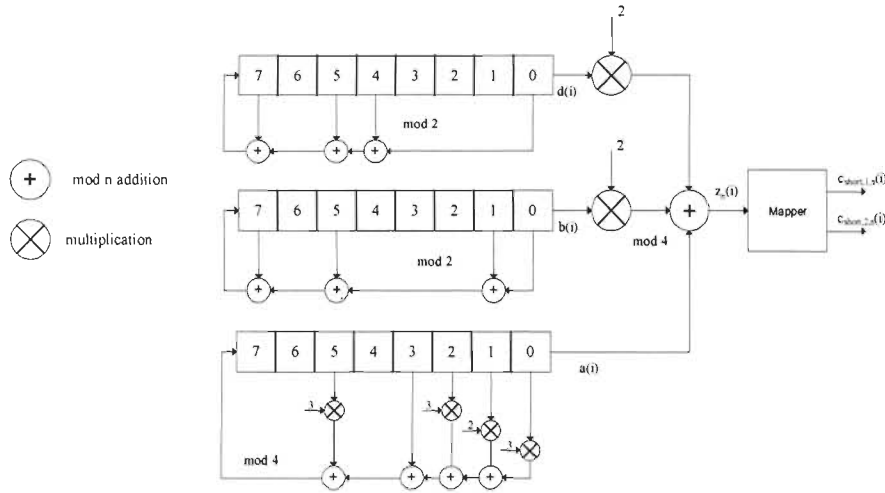


Figure 59 : Générateur des séquences des codes de brouillage courts (structure Fibonacci).

L'annexe B résume notre implémentation efficace sans multiplication d'un générateur de codes de brouillage que nous avons réalisé dans le cadre de ce travail de thèse.

A.3. Démodulation et désétalement de spectre

La figure 61 illustre le principe de démodulation et de désétalement basé sur un filtre apparié MF d'un récepteur. Le bloc « Décision » sert à décider de fournir des sorties dures ou douces au décodeur. Respectivement, le décodeur dur ou doux est utilisé. Dans cette structure, le désétalement est effectué par un filtre MF suivant par une opération de multiplication avec le code de brouillage. Le filtre MF apparie avec le code d'étalement d'un usager d'intérêt. Les codes d'étalement des usagers dans un système WCDMA sont orthogonaux.

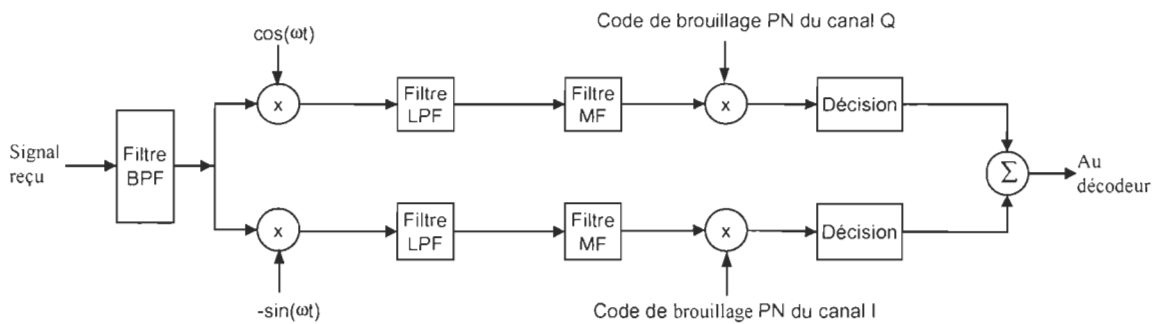


Figure 60 : Démodulation et désétalement de spectre basé sur un filtre apparié.

Tenant compte de la problématique déjà exposée au chapitre I (cf. I.1), notamment multi-trajet et interférences (ISI et MAI), un filtre MF seul n'est pas suffisant pour la reconstruction des données. Ainsi, la détection à usagers multiples a été proposée.

A.4. Débit de données

Le débit de données dépend du facteur d'étalement. Pour les codes courts de longueur 256, la valeur du facteur d'étalement est $2 \leq 2^k \leq 256$. Pour assurer les performances, la norme WCDMA exige un codage de canal, lequel ajoute les bits pour le codage. Par conséquent, dans une trame du canal *DPDCH* de 3,84 M brides, il y a les bits de données utiles et les bits pour des fins de codage de canal. Le tableau 29 montre les débits correspondants avec les facteurs OVSF de 64, 16, 8, et 4 [3].

Tableau 29 : Débits de données des facteurs OVSF types.

OVSF Débit	64	16	8	4
Débit DPDCH	60 kb/s	240 kb/s	480 kb/s	960 kb/s
Débit utile	12.2 kb/s	64 kb/s	144 kb/s	384 kb/s
Type de média	Sons	Données	Données	Données

A.5. Quelques caractéristiques radio de mesure

On entend, dans cette section, par caractéristiques radio de mesure comme les conditions de propagation telles que puissances et délais. Ces conditions de propagations varient selon les scénarios de communications (d'évanouissement). Ainsi, le tableau 30 résume les conditions de propagation pour des environnements d'évanouissement à trajets multiples. Dans ce tableau, les conditions de propagation varient selon la mobilité des transmetteurs [3].

Tableau 30 : Conditions de propagation pour des environnements d'évanouissement multi-trajet.

Case 1, speed 3km/h		Case 2, speed 3 km/h		Case 3, speed 120 km/h		Case 4, speed 3 km/h		* Case 5, speed 50 km/h		Case 6, speed 250 km/h	
Relative Delay [ns]	Relative mean Power [dB]	Relative Delay [ns]	Relative mean Power [dB]	Relative Delay [ns]	Relative mean Power [dB]	Relative Delay [ns]	Relative mean Power [dB]	Relative Delay [ns]	Relative mean Power [dB]	Relative Delay [ns]	Relative mean Power [dB]
0	0	0	0	0	0	0	0	0	0	0	0
976	-10	976	0	260	-3	976	0	976	-10	260	-3
		20000	0	521	-6					521	-6
				781	-9					781	-9

ANNEXE B

IMPLÉMENTATION DES SÉQUENCES DE CODES DE BROUILLAGE SANS MULTIPLICATION

Cette annexe vise à implémenter efficacement le générateur des codes de brouillage courts utilisés dans l'étalement de spectre des systèmes WCDMA.

B.1. Générateur de séquences de brouillage

Dans les systèmes 3G, chaque usager est différent par son code spécifique de 24 bits n qui est représenté en binaire comme $n_{23}n_{22} \dots n_0$. Défini par la norme 3GPP, la séquence de codes de brouillage courts est une séquence binaire à valeurs complexes et générée par trois phases [2]. La première phase consiste à générer une séquence quaternaire à valeurs réelles $z_n(i)$, où $i = 0, 1, \dots, 255$. La deuxième phase consiste à mapper la séquence quaternaire $z_n(i)$ à deux séquences binaires à valeurs réelles $c_{short,1,n}(i)$ et $c_{short,2,n}(i)$, où $i = 0, 1, \dots, 255$. La troisième phase consiste à générer la séquence binaire à valeurs complexes $C_{short,n}(i)$, où $i = 0, 1, \dots, 255$, à partir de deux séquences $c_{short,1,n}(i)$ et $c_{short,2,n}(i)$.

La séquence $z_n(i)$ est générée par l'addition à modulo-4 de trois séquences : la séquence quaternaire $a(i)$, la séquence binaire $b(i)$ et la séquence binaire $d(i)$. Elle est définie comme suit :

$$z_n(i) = \begin{cases} [a(i) + 2b(i) + 2d(i)] \text{ modulo } 4, i = 0, 1, \dots, 254 \\ z_n(0), i = 255 \end{cases} \quad (91)$$

La séquence quaternaire $a(i)$ est récursivement générée par le polynôme $g_0 = x^8 + x^5 + 3x^3 + x^2 + 2x + 1$ comme suit :

$$a(i) = \begin{cases} (2n_0 + 1) \text{ modulo } 4, i = 0 \\ (2n_0) \text{ modulo } 4, i = 1, 2, \dots, 7 \\ [3a(i-1) + a(i-5) + 3a(i-6) + 2a(i-7) + 3a(i-8)] \text{ modulo } 4, i = 8, 9, \dots, 254 \end{cases} \quad (92)$$

La séquence binaire $b(i)$ est récursivement générée par le polynôme $g_1 = x^8 + x^7 + x^5 + x + 1$ comme suit :

$$b(i) = \begin{cases} n_{8+i} \text{ modulo } 2, i = 0, 1, \dots, 7 \\ [n(i-1) + b(i-3) + b(i-7) + b(i-8)] \text{ modulo } 2, i = 8, 9, \dots, 255 \end{cases} \quad (93)$$

La séquence binaire $d(i)$ est récursivement générée par le polynôme $g_2(x) = x^8 + x^7 + x^5 + x^4 + 1$ comme suit :

$$d(i) = \begin{cases} n_{16+i} \text{ modulo } 2, i = 0, 1, \dots, 7 \\ [d(i-1) + d(i-3) + d(i-4) + d(i-8)] \text{ modulo } 2, i = 8, 9, \dots, 254 \end{cases} \quad (94)$$

Le *mapping* de la séquence quaternaire à valeurs complexes $z_n(i)$ aux deux séquences binaires à valeurs réelles $c_{short,1,n}(i)$ et $c_{short,2,n}(i)$ est défini comme dans le tableau 31.

La séquence binaire de brouillage à valeurs complexes $C_{short,n}(i)$ est définie comme suit :

$$C_{short,n}(i) = C_{short,1,n}(i \bmod 256)(1 + (-j)^i C_{short,2,n}(2 \lfloor (i \bmod 256)/2 \rfloor)) \quad (95)$$

où $\lfloor \bullet \rfloor$ dénote l'arrondissement à un entier le plus près et le plus petit.

Tableau 31 : *Mapping* de la sequence $z_n(i)$ aux séquences $c_{short,1,n}(i)$ et $c_{short,2,n}(i)$.

$z_n(i)$	$c_{short,1,n}(i)$	$c_{short,2,n}(i)$
0	+1	+1
1	-1	+1
2	-1	-1
3	+1	-1

B.2. Implémentation matérielle sans multiplication du générateur des séquences de codes de brouillage courts

La norme 3GPP a proposé une implémentation du générateur des codes de brouillage courts comme présentée à la figure 60 de l'annexe A [2]. Il s'agit d'une structure Fibonacci [61]. Alternativement, on peut implémenter avec une structure Galois comme présentée à la figure 62.

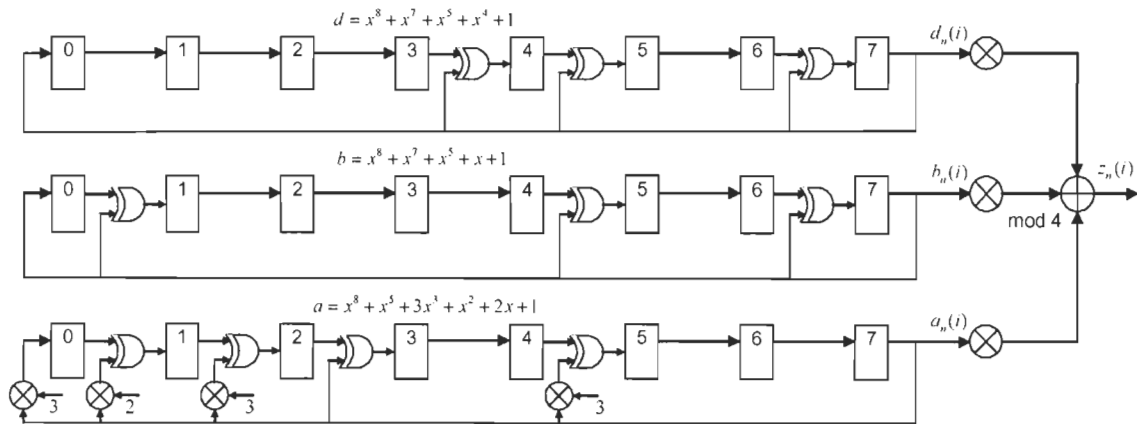


Figure 61 : Structure Galois du générateur des codes de brouillage courts.

L'additionneur à modulo-4 constitue un module important pour implémenter le registre à décalage défini par le polynôme $a(i)$ dans la figure 60 (structure Fibonacci) ou la figure 62 (structure Galois). Il peut être implémenté sans multiplication comme présentée à la figure 63.

La figure 63.a illustre le principe d'un modulo-4. En effet, le résultat d'une opération modulo-4 sur un nombre $a_{n-1}...a_1a_0$ est simplement une sélection de deux bits les moins significatifs (a_1a_0). La séquence quaternaire $a(i)$ peut être codée par deux bits. En plus, on peut constater que l'additionneur à modulo-4 dans la structure du générateur des codes de brouillage s'effectue avec un entier de 3 et un entier de 2. Ces derniers peuvent être aussi codés par deux bits comme '11' et '10' respectivement. Ainsi, l'additionneur à modulo-4 avec 3 et 2 est implémenté comme présentée à la figure 63.b et figure 63.c respectivement. En appliquant la distribué de l'addition, l'additionneur à modulo-4 avec 3 et 2 peut être réalisé sans multiplication.

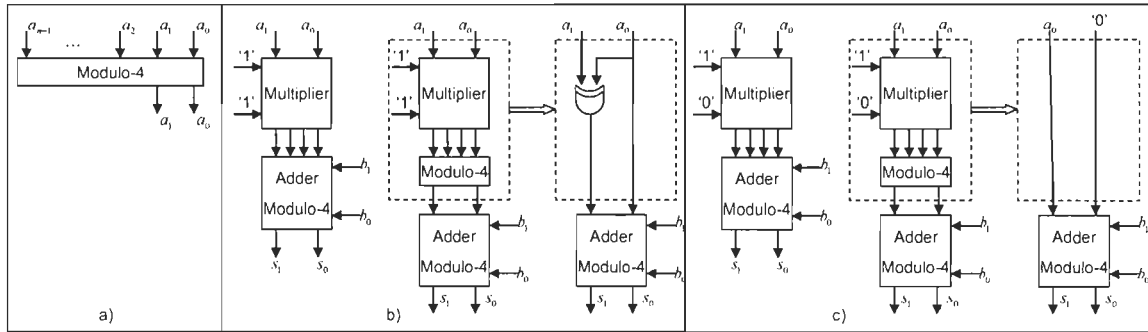


Figure 62 : Implémentation du modulo-4 dans la structure du générateur des codes de brouillage courts.

La figure 64 présente la génération de la séquence $C_{short,n}(i)$ à partir de deux séquences $c_{short,1,n}(i)$ et $c_{short,2,n}(i)$. L'équation (95) est réalisée par une structure matérielle comme à la figure 64.a. Dans cette structure, la séquence est implémentée par les portes logiques ou-exclusive et multiplexeurs comme présentée à la figure 64.b.

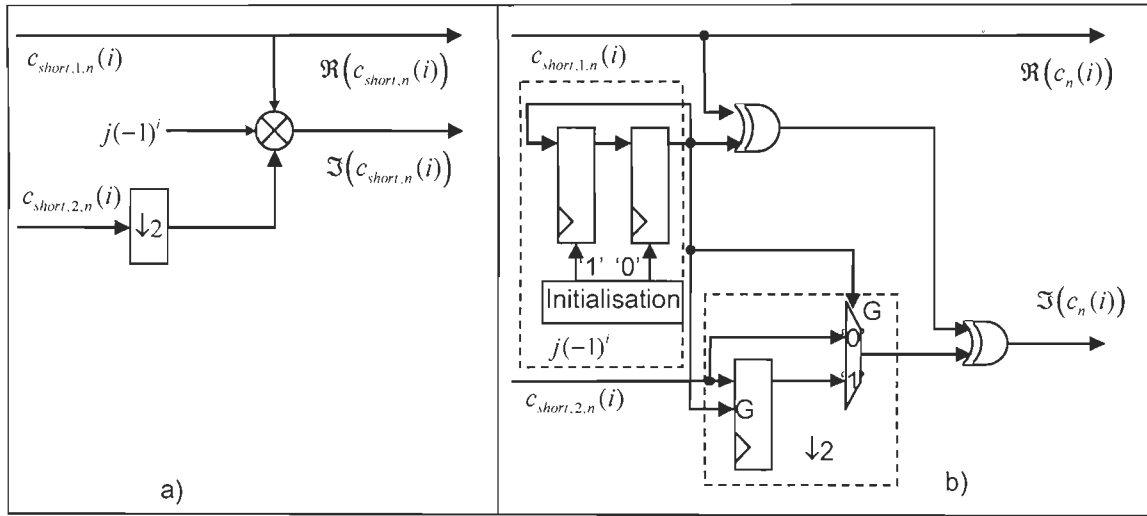


Figure 63 : Génération de la séquence $C_{short,n}(i)$.

INDEX

- 2G, 2
- 3G, 1
- Algorithmic CTM, 52
- Approche conventionnelle, 111
- Approche unifiée, 117
- Bloc Détection, 58
- Bloc Signature, 56
- canal physique, 153
- caractéristique radio, 159
- CDMA, 1, 11
- cdma2000, 1
- CF-MUD, 3, 54, 88, 140
- Complexité, 32
- Débit, 158
- Décorrélateur, 26
- Démodulation, 158
- désétalement de spectre, 158
- Détection à usagers multiples multi-débit, 33
- Doppler, 5
- DS-CDMA, 1, 16
- étalement de spectre, 155
- FDD, 152
- Filtre apparié, 22
- Flot de conception, 109
- HRD-CF-MUD, 89
- HRD-Sign-MUD, 98
- IS-95A, 2
- ISI, 2
- liaison montante (*uplink*), 152
- LRD-CF-MUD, 89
- LRD-Sign-MUD, 97
- MAI, 2, 4, 14
- mémoires, 66
- Méthode HRD, 41
- Méthode LRD, 37
- Méthode successive, 43
- Modèle de système, 16, 34
- Modulation, 155
- MUD, 2, 20
- multi-débit, 88
- Multiplexage temporel, 65
- near-far*, 5
- Objet fi, 51
- proche-loin, 5
- Rake, 5
- Récepteur MMSE, 26
- Récepteur optimal, 25
- Récepteur PIC, 29
- Récepteur Rake, 23
- Récepteur SIC, 27
- séquences brouillards, 160
- SIC, 5
- SSMA, 11
- SystemC, 51
- TD-SCDMA, 1
- trajets multiples, 3
- virgule fixe, 50, 127
- WCDMA, 1, 11, 152
- ZF, 26